JPRS L/10309

8 February 1982

USSR Report

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 2/82)

FBIS FOREIGN BROADCAST INFORMATION SERVICE

NOTE

JPRS publications contain information primarily from foreign newspapers, periodicals and books, but also from news agency transmissions and broadcasts. Materials from foreign-language sources are translated; those from English-language sources are transcribed or reprinted, with the original phrasing and other characteristics retained.

Headlines, editorial reports, and material enclosed in brackets [] are supplied by JPRS. Processing indicators such as [Text] or [Excerpt] in the first line of each item, or following the last line of a brief, indicate how the original information was processed. Where no processing indicator is given, the information was summarized or extracted.

Unfamiliar names rendered phonetically or transliterated are enclosed in parentheses. Words or names preceded by a question mark and enclosed in parentheses were not clear in the original but have been supplied as appropriate in context. Other unattributed parenthetical notes within the body of an item originate with the source. Times within items are as given by source.

The contents of this publication in no way represent the policies, views or attitudes of the U.S. Government.

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500030018-4

JPRS L/10309

8 February 1982

USSR REPORT

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 2/82)

CONTENTS

GENERAL

UDC 621.391

VARIATION OF FEEDING VIDEO INFORMATION TO YES COMPUTERS

Kiev AVTOMATIKA in Russian No 5, Sep-Oct 81 (manuscript received 19 Mar 81) pp 76-78

[Article by V. Ye. Reutskiy]

[Text] There are many problems involving processing video information in various fields of science and technology. The growing volume of this information and the heightened requirements for precision and speed of solutions led to intensive development of means and methods of automating the processing of video information. Digital processing methods using general-purpose computers and specialized processors play the leading role in this. This leads to the necessity of developing new input units, on the one hand, and a desire to use input units already included in the computer equipment on the other.

Feeding video information to computers requires a unit to match the information and physical characteristics of the television camera signals and the signals used in the input/output interface of the particular computer. In this case equipment expenditures to realize the block to match the characteristics of the signals of the input/output interface of the YeS [Unified System] computer are several times as much as expenditures to realize the matching unit for the characteristics of television camera signals.

For this reason there will unquestionably be interest in an experiment conducted with a simple hardware connection of an industrial television camera to a YeS computer through a low-power input unit with minimum expenditure of time and resources.

The variation of feeding video information that was developed consists of a television camera, a matching element, and a YeS-6022 unit (the standard punched tape data input unit for YeS computers) in which the photoreader is switched off.

A VZOR television camera was used to convert visual information into an electrical analog signal. This is a small semiconductor camera with a power supply of 12 volts (in principle any standard television camera can be used). For complete autonomy the camera also has a synchronous generator (consisting of a synchronous pulse generator, a frequency divider, and control signal shapers) which controls the work of the television camera, analog/digital convertor, and control block.

The frequency of the master clock (generator) was selected at around 936 kilohertz on the basis of preliminary tests of the multiplex channel of a YeS computer to which was connected a YeS-6022 unit with a single-field data input speed on the order of 260-280 kilobytes, based on the real speed and necessity of receiving from one generator synchronous pulses to control the television camera and strobing pulses for the analog/digital convertor and control block. The possibility of completely filling the byte being transmitted to the YeS computer was employed to increase the speed of data input. A four-gradation system of defining the brightness of each point in the video information was adopted for this purpose, and the bit format of the analog/digital convertor was defined in two ranks. After the information from four points is packed in one byte, the data input speed is 234 kilobytes.

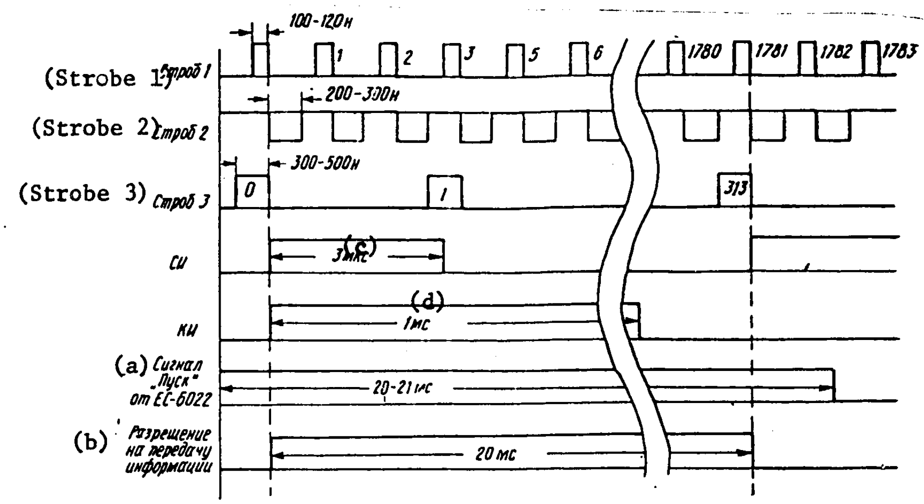
The precision requirements for the analog/digital convertor are fairly low (four gradations with a television camera signal on the order of 0.5 volts), but the speed requirement is quite important. Therefore, a double-rank (simplified) variation of a high-speed analog/digital convertor was used. This will make it possible in the future to increase the speed to 8-10 megahertz because it is determined chiefly by the operating speed of the comparators used in the convertor.

The television camera and analog/digital convertor work asynchronously with the computer, but information is transmitted only when the signal of the computer processor which authorizes transmission of information from the YeS-6022 coincides with the beginning of transmission of a frame by the television camera. After the frame is transmitted the YeS-6022 unit goes into a waiting mode and the next frame is fed following a program request by the computer processor.

The entire television camera signal matching block, the synchronous generator, and the signal shapers are housed in a standard YeS TEZ [card] which is installed in the YeS-6022. When the television camera is connected in, it is necessary to remove the input/output signal shapers of the photoreader (TEZ 0071, place 5A43 and TEZ 0072, place 5A44) from the unit and insert the newly developed TEZ in place 5A43. The necessary connection between place 5A43 and the free terminals is installed, which does not hinder work of the YeS-6022 with the photoreader.

According to the system adopted of classifying brightness by four levels, the frequency of the master clock is four times the frequency of data transmission to the computer, and the synchronous generator issues control signals that are rigidly interrelated by multiples: strobe 1 — strobing signal of the comparators of the analog/digital convertor — 936 kilohertz; strobe 2 — strobing signal of the assembly for packaging in a byte — 936 kilohertz; strobe 3 — signal for transmission of a byte of information to the YeS-6022 — 235 kilohertz; synchronous signal of the line frequency for the television camera — 16,625 hertz; synchronous signal of the frame frequency for the television camera — 50 hertz.

The figure below shows the time scheme of the work of the synchronous generator. It is built with integrated microcircuits and housed on the same board with the other elements of the matching TEZ.



Time Diagram of the Work of the Matching Unit

Key: (a) Launch Signal from YeS-6022;
(b) Authorization To Transmit Information;
(c) 3 Microseconds;
(d) 1 Milliseconds ["MC" = Millisecond].

Because the computer and the television camera work asynchronously, an additional element was put in the control scheme to authorize transmission of data from the television camera. It authorizes transmission only from the initial moment of image scanning by the television camera. The end of the transmission of a frame is determined by program means (by the number of transmitted bytes of the array of data).

As noted above, the speed of information input is limited by the actual speed of the unit being used and the channel of the YeS computer. Therefore, with the above indicated frequency of data transmission to the YeS computer (234 kilobytes), the analog/digital computer strobes and converts to digital form 60 four-gradation points or (after packaging) 15 bytes of information per one television line where the total number of lines in a frame is 313. Thus, one television frame contains 4,695 bytes or 18,784-gradation points in digital form.

With a standard line length of 64 microseconds the distance between adjacent strobing points is about 1 microsecond in time. Therefore, all points that occupy less than 1 microsecond during line strobing may be lost. This must be taken into account when feeding data from the television camera.

The image fed to the computer may be printed out by program means for convenient observation and checking, although this does involve difficulties with observing the television format during output. In the particular case every fourth line of the television frame was outputted to a digital printer; in this case 78 lines with 60 points to a line were printed out and separated by gaps to approximate the television format.

11,176
CSO: 1863/48

UDC 681.322.01

SYNTHESIS OF PARALLEL MICROPROGRAMMING STRUCTURES

Kiev KIBERNETIKA in Russian No 5, Sep-Oct 81
(manuscript received 23 Aug 77) pp 48-54

[Article by Ol'ga Leonidovna Bandman, Sergey Vladimirovich Piskunov and Stanislav Nikolayevich Sergeyev]

[Excerpts] The organization of effective computations in uniform structures [1-3] is connected with providing structural and functional correspondence between the algorithm and its performing unit. This correspondence is usually achieved by means of a selection of cell functions and their connections and constitutes a problem in synthesizing uniform structures. The solution of such a problem for a class of devices with a multiple flow of commands and a multiple flow of data is proposed in the article. The suggested synthesis method is based on a microprogrammed presentation of algorithms which allows interpretation to the network from automatic units. The microprogramming concept, expanded for organization of the computational process in large groups of low-power computers, is called parallel microprogramming.

A cellular mass, which is a terminal combination of named cells into which symbols are inserted from some terminal alphabet of conditions, serves as the conversion objective in parallel microprogramming. A substitution microcommand having a left and a right component is the primary conversion of the cellular mass. Execution of the microcommand is conducted simultaneously over all cells of the mass. A mass of cells is computed for each cell in accordance with the left component of the microcommand. If the obtained mass is included in the processed cellular mass, some of its component is replaced by the cellular mass corresponding to the right component.

The combination of microcommands recorded in random order is called microprogramming. All microprogramming microcommands are executed simultaneously over all cells of the mass. Microprogramming construction is based on algorithms of parallel substitutions [4, 5] which are based on the concepts of a cellular automatic unit [6] and of a normal algorithm [7].

Any methods of synthesis can be used for obtaining a logical structure of the network's primary automatic unit. For example, each primary automatic unit becomes microprogrammed, containing within itself the entire microprogram. A block diagram of the primary automatic unit is depicted in illustration 5.a. Equipment performance is possible when using special microprogrammed large integrated circuits (BIS), constant memory units (PZU), associated memory units (ZU) and programmed logic matrices. Replacing the microprogram provides the possibility of adjusting the network to that or another algorithm. Networks of this type are called microprogrammed cellular structures. It is obvious that the parallel microprograms performed in them are limited by the memory capacity of the primary automatic units, and therefore such an approach is acceptable for performing small (according to the number of microcommands) algorithms.

The second approach consisting of a microprogram memory which is repeated in each automatic unit is carried out within the limits of the network (illustration 5.b.). The remaining network is called the data memory. The primary automatic units in it are simple enough: they contain a memory for storing the alphabet symbols, a comparison unit and a multiplexer [4]. The microcommands can be fed into the data memory with a different degree of parallel in time: all simultaneously, successively by microcommands and even successively by symbols. However, a marginal parallel in space takes place in all cases. Performances of this category are called homogeneous computers [4].

It is obvious that computers with limited parallel both in time and in space have the greatest output. A structure is proposed in reference 12 in which during performance of the primary automatic units, a combination of frequent and spatial coding of microcommand symbols and the use of the construction principles of multi-stable elements provide a simultaneous feeding of hundreds of microcommands to each automatic unit fo the data memory. At the same time, homogeneous computers with a consistent delivery of microcommands to the data memory show great practical interest, inasmuch as their performance is possible from existing microprocessing sets [13].

Two approaches can be used in composing algorithms for equipment interpretation. In the first case, a control unit is proposed which stores the microprogram F_s and delivers control signals to the microprogram structures performing $F'_1, \ldots\ldots, F'_q$ [14]. In the second case, the memory is divided into units corresponding to the data for $F'_1, \ldots, F_q$, and the microprogram memory contains the microprogram comprising F.

Bibliography

1. Evreinov, E. V. and Prangishvili, I. V., "Digital Automatic Control Units With an Adjusted Structure," Moscow, Energiya, 1974, 239 pages.

2. Bandman, O. L., Evreinov, E. V., Korneyev, V. V. and Khoroshevskiy, V. G., "Homogeneous Computer Systems on the Basis of Microprocessing BIS [Large Integrated Circuits], in the collection "Problems of Theory and Construction of Computer Systems," No 70, 1977, pp 3-28.

3. "Homogeneous Microelectronic Associated Processors," edited by I. V. Prangishvili, Moscow, Sovetskoye radio, 1973, 280 pages.

4. Kornev, Yu. N., Piskunov, S. V. and Sergeyev, S. N., "Algorithms of Generalized Substitutions and Their Interpretation by Networks of Automatic Devices and Homogeneous Computers," IZVESTIYA AKADEMII NAUK SSSR: TEKHNICHESKAYA KIBERNETIKA, No 6, 1971, pp 131-142.

5. Kornev, Yu. N., Piskunov, S. V. and Sergeyev, S. N., "Problems of Constructing Algorithms of Generalized Substitutions in a Separate Context," VYCHISLITEL'NYVE SISTEMY, No 47, 1971, pp 117-119.

6. Yamada, H. and Amoroso, S., "Tesselation Automata," INFORM. AND CONTROL., Vol 14, No 3, 1969, pp 299-317.

7. Markov, A. A., "Theory of Algorithms," TRUDY MATEM. INSTITUTA AKADEMII NAUK SSSR, No 42, 1954, p 375.

8. Peterson, I. L., "Petri Nets," ACM COMPUT. SURVEYS, Vol 9, No 3, 1977, pp 233-252.

9. Lipton, R. I., Snyder, L. and Zalzstein, Y. A., "Comparative Study of Models of Parallel Computation," 15th Annual Simp. Switching and Autom., 1974, pp 145-155.

10. Baranov, S. I., "Synthesis of Microprogrammed Automatic Units," Leningrad, Energiya, 1974, 216 pages.

11. Anishev, P. A., "Concerning the Determinability of Parallel Chart-Diagrams," in the collection "Problems of Theory and Construction of Computer Systems," No 75, 1978, pp 40-52.

12. Author Certificate 664168 (USSR), "Computational Homogeneous Structure," Kornev, Yu. N. and Piskunov, S. V., Published in B.I., No 19, 1979.

13. Sergeyev, S. N., "Performance of Algorithms of Parallel Substitutions in Microprocessing Systems," in the collection "Problems of Theory and Construction of Computer Systems," No 73, 1978, pp 25-39.

14. Bandman, O. L., "Synthesis of Asynchronous Microprogrammed Control by Computer Processes," KIBERNETIKA, No 1, 1980, pp 42-47.

9889
CSO: 1863/52

UDC 518.9

VERIFICATION OF MODEL CONFIGURATIONS FOR PULSED RADIO ENGINEERING SYSTEMS

Kiev KIBERNETIKA in Russian No 5, Sep-Oct 81
(manuscript received 3 Jun 80) pp 40-47

[Article by Yuriy Anatol'yevich Belov, Vladimir Leonidovich Makarov, Viktor Gennadiyevich Shelepov and Vladimir Borisovich Shul'zhenko]

[Excerpts] When examining models like operators transforming some given input data into definite output data, the question always arises whether the process of handling the flow of incoming information corresponds to that function which the model must perform. In other words, how to be convinced that the correct results are being received in any of the processes which can be simulated in accordance with the model data.

The question of adequacy of the mathematical models is one of the basic, decisive and inevitable questions of modeling.

Considering the complexity of checking such compliance for models of actual or projected radio engineering systems (RTS), it is possible to formulate a more simple task--to check whether the model (operator) is presenting data from the field of permissible incoming values to data of the field of permissible output values.

A number of well-founded methods of checking program accuracy was worked out for solving the immediate problem in programming: Floyd's inductive supposition method [1], the hierarchical structuralization method [2], Hoare's method [3] and others.

The purpose of present work is to develop an inductive supposition method of checking program accuracy for a new and important, in a practical respect, class of mathematical objectives--model configurations.

It should be mentioned that strict proof of accuracy in constructing a mathematical model and the corresponding simulation algorithm according to an assigned physical description (structural configuration) of the IIS [Information measuring System] is an important procedure which allows, generally speaking, one to ascertain the inadequacy of the physical model to the actual process and, therefore, to localize operations for correcting the model on the level of physical presentations alone.

Bibliography

1. Manna, C., "Mathematical Theory of Computation," New York, McGraw-Hill Book Co., 1974, 226 pages.

2. Robinson, L. and Levitt, K., "Proof Techniques for Hierarchically Structured Programs," CURRENT TRENDS IN PROGRAMMING METHODOLOGY, No 11, 1979, pp 57-73.

3. Hoare, C. A. R., "An Axiomatic Basis of Computer Programming," CACM, Vol 12, No 10, 1969, pp 112-135.

4. Kotov, V. E., "Introduction to the Theory of Program Configurations," Moscow, Nauka, 1978, 257 pages.

5. Leonov, A. I. and Fomichev, K. I., "Monopulse Radar," Moscow, Sovetskoye radio, 1970, 392 pages.

6. Leonov, A. I., Vasenev, V. N. and Gaydukov, Yu. I., "Radar Simulation," Moscow, Sovetskoye radio, 1979, 264 pages.

7. Kuz'min, S. Z., "Digital Processing of Radar Information," Moscow, Sovetskoye radio, 1967, 399 pages.

8. King, Dzh., "Checking Compiler," in the book "Aids To Managing Large Systems," Moscow, 1977, pp 23-41.

9. Anisimov, A. V., Belov, Yu. A., Lyashko, I. I. and Makarov, V. L., "The Adequancy of Mathematical Modeling of a Complex Information Measuring System," DOKLADY AKADEMII NAUK SSSR, Vol 240, No 2, 1978, pp 202-206.

10. Belov, Yu. A., Makarov, V. L., Shelepov, V. G. and Shul'zhenko, V. B., "One Approach to Checking Adequacy of the Block Diagram of a Functional Algorithm for the Structural Configuration of a Pulsed Information Measuring System," DOKLADY AKADEMII NAUK SSSR, Vol 255, No 1, 1980, pp 36-40.

9889
CSO: 1863/52

EFFICIENT SERVICING OF COMPUTER EQUIPMENT REVIEWED

Moscow TEKHNICHESKIYE SREDSTVA OBRABOTKI INFORMATSII in Russian 1981 (signed to press 17 Feb 81) pp 307-318

[Selections from chapter 9 of book "Data Processing Equipment", by Vasiliy Nikolayevich Kriushin, Nikolay Matveyevich Surin, Valeriy Pavlovich Chuprikov and Nina Grigor'yevna Chernyak, Izdatel'stvo "Finansy i statistika", 12,000 copies, 320 pages]

[Excerpts] Table 9.1. Technical Servicing Norms for Computer Keypunch, Electromechanical, and Electronic Keyboard Machines by One Worker at Computing (Information-Computing) Centers, Information-Computing Stations, and Machine-Accounting Stations of the System of the USSR Central Statistical Administration.

Types and Models of Machines	Number of Machines Serviced by One Worker in a Month	Average Number of Hours To Service One Machine for a Month
Machines for Mathematical Processing of Data Recorded on Punchcards		
The T-5M and T-5MV Tabulators	5	34.9
The TA80-1 Tabulators	3	58.2
The VP-2 and VP-3 Electronic Computing Attachments to Tabulators	9	19.4
Machines for Ordering Arrays of Punchcards		
The SE80-3 and SE80-3/1M Electronic Sorters	6	29.1
The S80(45)-5M and S80(45)-7 Sorters	10	17.5
Collaters (RPM's)	7	24.9
Machines for Preparing Punchcards		
The PD 45-2 and PD 45-2/1M Keypunches	12	14.5
The P80-6, P80-6/1M, and Zoyemtron-415ts Keypunches	9	19.4

Types and Models of Machines	Number of Machines Serviced by One Worker in a Month	Average Number of Houses To Service One Machine for a Month
Machines for Preparing Punchcards		
The PA80-2, PA80-2/1M, PA80-2/2M, PA80-2/3M, and Zoyemtron-415a Keypunches	6	29.1
Summary Punches of the PI80(45)-U Type	11	15.9
Reproducing Punches of the PR80(45)-U Type	9	19.4
The K80(45)-6, K80(45)-6/1M, and Zoyemtron-425ts Verifiers	10	17.5
The KA80-2, KA80-2/1M, KA80-2/2M, KA80-2/3M, and Zoyemtron-425a Verifiers	7	24.9
The RMK80/45 and RMA-80 Decoding Machines	8	21.8
Keyboard Machines		
The SDV-107, SDV-108, SDK-133, AYeS, AYeSVye, Askota-110, -112, -114, -117, and -314, and Other Adding Machines	30	5.8
The VMA-2, VK-2, SAR, KYeL, and Other Electro-Mechanical Keyboard Computing Machines	25	6.9
The Elka-22, Zoyemtron-220, Elektronika, and Other Electronic Keyboard Computing Machines with Discrete Elements	30	5.8
The Iskra-108, -111, and -122, Elka-43 and Elka-50, and Other Electronic Keyboard Computing Machines with Integrated Circuits	50	3.5
The VA-345M, FM-346, FMR, FMYe, and FMYeL Invoice Machines	9	19.4
The Zoyemtron-381 and 382, Iskra-23 and 522, and EFM-446 Electronic Invoice Machine	12	14.5
The Zoyemtron-381, -383, -384, and -385, Iskra-2302 and -534 Electronic Invoice Machines with Punch Attachments	9	19.4
The Askota-170 Bookkeeping Machines	9	19.4
The TM-20 Electronic Multiplier Attachments	27	6.5
The Askota-170/55 Bookkeeping Machines with Tape Punch Attachments	8	21.8

Technical servicing and repair of computer equipment is done by enterprises of Soyuzschettekhnika (All-Union Production-Technical Association for Technical Servicing and Repair of Computer Equipment of the USSR Central Statistical Administration) on the basis of contracts concluded between the manufacturing enterprise and the client.

Soyuzschettekhnika repair enterprises can do technical servicing, current (small-scale) repair, medium repair (1 and 2), and capital repair (1 and 2).

With the approval of the USSR State Committee for Prices a price list for wholesale prices for all types of repair work and technical servicing of the computer equipment of the USSR Central Statistical Administration was ratified and put into effect as of 1 January 1977. This price list envisioned separate wholesale prices for different types of repair, technical servicing, and other jobs (launching machines into operation, storage and packaging, repair of electrical motors, repair and replacement of particular assemblies). Where there is a contract for technical servicing, current (small-scale) repair, installation, and launching newly received equipment in operation are done by Soyuzschettekhnika enterprises free of charge.

Let us review the planning for repair of computer equipment using the example of organizations of the USSR Central Statistical Administration. The established rules for planning repair of computer equipment within the system of the USSR Central Statistical Administration envision that rayon and city computing centers (and machine accounting stations) will submit information on the technical condition and number of hours worked by each machine in the past year and plans for the machine's workload in the current year to the computing centers (machine accounting stations) of the oblast statistical administrations and the appropriate repair enterprises each year (based on condition as of 1 January). On the basis of these data the repair enterprise makes up a technical passport for each machine, sets up a card file, and plans and maintains records of repair work.

The rayon and city computing centers (machine accounting stations) submit a plan of machine repair for the coming year to the computing centers (machine accounting stations) of the oblast statistical administrations each year, before 1 February of the current year, according to the number of hours worked by each machine. This plan indicates repair enterprises by service zones using the following form:

Form No 9.1

Plan for Repair of Computer Equipment in 198__

____

(Name of Computing Center or Machine Accounting Station)

For the Service Zone of ______________________________

(Name of Repair Enterprise)

Machine Name	Factory Number	Number of Items	Included in that		
			Capital Repair	Medium Repair	Current Repair

Each year the computing centers (machine accounting stations) of the oblast statistical administrations compile a summary request (plan) in the same form showing their repair needs for the coming year. This is submitted to the Main Administration of Computer Work (Glavmekhschet) of the republic central statistical administration before April of the current year. This administration then sends the summary request for the republic central central statistical administration to Glavmekhschet of the USSR Central Statistical Administration before 1 June of the current year. By 1 July of the same year Glavmekhschet of the USSR Central Statistical Administration submits information on the computer equipment reapir needs of the system of the USSR Central Statistical Administration to the Soyuzschettekhnika Association. Soyuzchettekhnika compiles a summary plan of computer equipment repair and brings it to the attention of the repair enterprises.

At the large computing centers (machine accounting stations) where medium and current (small-scale) repair is done by in-house personnel, the work planning system envisions compiling annual plans of machine repair which list all machines subject to repair and indicate the type of repair and the time it will take for each machine. In addition to compiling the annual repair plan they work out a repair schedule for each month which envisions planned withdrawal of machines from work positions. The schedule is usually drawn up on the basis of the volume of repair work in the current month and the calendar amount of working time of workers engaged in repair. The schedule is used for operational monitoring and accounts related to calculating expenditures for repair. Detailed trouble lists are compiled before performing each type of repair

Repair workshops which have the necessary amounts of furnishings, bench equipment, tools, and spare parts are organized to do repair work at the computing centers (machine accounting stations). In the repair workshops the work of service personnel is usually organized on the brigade principle: each brigade is assigned to a certain group of machines. In addition, a definite worker is responsible for repairing each machine.

The Organization of Technical Servicing

Technical servicing of computers involves a set of organizational-technical measures which must be done to maintain the operating reliability of machines within required parameters. These activities include: selection of appropriate service personnel; acquisition of hardware and software for diagnosing malfunctions; supplying spare parts, tools, and accessories for the machines; supplying service apparatus for testing external units; supplying special furnishings and auxiliary equipment for operating and repairing computers.

Optimal organization of the technical servicing of computers is a major task in using them efficiently. There are various types of technical servicing of computers: individual, group, and centralized.

With individual servicing each computer is provided with a complete set of service apparatus, spare parts, tools, accessories, and appropriate service personnel. Individual service facilities include the following: apparatus for monitoring the basic elements of the computer and power supply; monitoring and adjusting apparatus for autonomous checks of individual units of computer hardware; a set of test programs; tools and repair accessories; auxiliary equipment, attachments, and special furnishings for storing the property of the computing center.

With group servicing several computers concentrated at a single computing center are serviced by in-house personnel. The structure of group service is the same as for individual service, but it is supplemented for other computers.

With centralized computer servicing the service apparatus, spare parts, tools, accessories, and service personnel are significantly reduced, but with this form of servicing the time required to restore computers depends significantly on the operational features of the work of centralized servicing points, in particular how far they are from the computing centers.

The type of technical servicing is determined by the user depending on the place where the computer is installed and the range of jobs it does.

The number of engineering-technical personnel needed to service computers depends on the type of technical servicing and the mode of operation of the computer. Thus, for individual servicing of one YeS-1020 computer which is used in three shifts, the following engineering-technical staff is recommended:

Machine Chief	1
Shift Chiefs (Senior Engineers)	3
Shift Engineers (Electrical Engineers and Electromechanical Engineers)	3
Senior Electricians	2
Electricians	2
Precision Machinery Mechanic	1
Total	12

Scheduled preventive maintenance work involves a set of measures aimed at keeping computer units in working condition and preventing breakdowns and failures during their operation.

The period of scheduled preventive work is an essential and continuing stage in keeping computers in working condition. Reducing the time that this work takes increases the usable work time of the computer, that is, the time during which the machine is engaged in problem-solving or debugging programs.

The volume of scheduled preventive maintenance work depends on the technical condition of the computer units and the qualifications of engineering-technical personnel. The length and periodicity of this work are determined by the manufacturing plants in the appropriate operating instructions.

The essential feature of preventive work is the following: when a machine is being prepared for problem-solving the working condition of the machine itself as well as its blocks and individual elements must be tested using specially prepared problems or test-programs with known answers. In case of an error the programs envision that the machine will halt automatically and a description of the nature of the trouble will be printed out on the typewriter or a signal will appear on the control console. The test program is used to check not only the condition of the computer, but also whether the program to solve the particular specific problem has been written correctly.

Experience operating contemporary computers shows that it is necessary to observe each machine systematically, clean its blocks and assemblies, test the work of individual units, and replace wornout parts. In addition to identifying and eliminating ordinary malfunctions, servicing computers involves daily, weekly (biweekly), monthly, and semiannual (annual) scheduled preventive maintenance.

The following periodicity and length of scheduled preventive maintenance work is recommended for YeS [Unified System] computers:

Table 9.4

Computer Name	Periodicity of Preventive Work	Number of Hours
YeS-1020	Daily	1
	Weekly	4
	Monthly	16
	Annual	72
YeS-1030	Daily	1
	Biweekly	4
	Monthly	8
	Semiannual	72
YeS-1050	Daily	1
	Biweekly	4
	Monthly	8
	Semiannual	72

Daily preventive maintenance includes an external inspection of the condition of the machine units and power sources; testing the machine with monitoring tests; eliminating malfunctions when there are deviations from technical norms; checking, cleaning, and adjusting external units.

Weekly (biweekly) preventive maintenance envisions a set of weekly (biweekly) activities: testing the reliability of mechanical fastenings; testing the ventilation and power supply system; checking the work of the units with a preventive alteration of the supply voltage of ±5 percent.

Monthly preventive work envisions cleaning and lubricating the assemblies, mechanisms, and power blocks and testing the functioning of the computer hardware using diagnostic tests with an alteration of ±5 percent in supply sources.

Annual (semiannual) preventive work includes the same jobs that are done in monthly preventive work as well as dismantling, cleaning, and lubricating all mechanical assemblies of external units with adjustment or replacement of parts at the same time. In addition, the cables and supply lines are inspected. The instructions on operating individual units which are attached to the machine by the manufacturing plant give detailed descriptions of preventive work. They also indicate possible malfunctions in the machines.

Careful scheduled preventive maintenance work significantly reduces the appearance of random malfunctions in machines. But finding and eliminating malfunctions quickly depends in large part on the experience and qualifications of the engineering-technical personnel who are operating the machines.

Types of Malfunctions and Methods of Detecting Them

Malfunctions may occur in computers for various reasons. Among them are concealed production defects, violations of operating rules, and external influences (blows, vibrations, overheating, and the like). Malfunctions occurring for these reasons generally come during the period of experimental operation.

During working operations most malfunctions are related to failures of integrated circuits and semiconductor instruments. Experience operating the machines shows that the cause of this is instability of the voltage in the machine circuits. Failures in computers often occur because resistors go out, despite their high reliability. More than 50 percent of all failures occur because the contacts that connect the current-conducting element with the outlets are broken or disrupted. Failures also occur as the result of rupture of the dielectric in the capacitors.

Trouble-free operation of the semiconductor instruments depends on how well they are manufactured. These instruments are sensitive to overloads of current and voltage and therefore above-limit power supply regimes must be avoided if they are to be used for a long time. Semiconductor instruments are also subject to the influence of heat. An increased temperature causes change in the parameters of the element and can result in failure of the computer.

To make the search for malfunctioning elements easier each element in the machine has an address which gives the number of the unit, the frame, console, and place which the particular element occupies. These coordinates are shown in the structural and schematic diagrams by which the search for malfunctions is conducted.

The machines are supplied with service apparatus, including an oscillograph, an electrical measurement instrument, a stand for testing TEZ's [cards] and a stand for testing power supply blocks, among others, to determine the operating stability of the particular elements. This apparatus plays a significant part in the search for machine malfunctions. However, the automatic monitoring equipment is primary. It makes it possible to check whether the machine is operating correctly and, in many cases, to correct errors that are detected. This includes both hardware and software for automatic monitoring.

The software includes monitoring programs for test problems which are run before performing the main problem. When the test problems are solved correctly the probability of correct work by the machine in solving the main problem is determined. During solution of the main problem monitoring procedures are carried out in the form of a double check of the same program.

The hardware includes monitoring equipment that operates independently of the program. For example, the YeS-1020 processor uses hardware monitoring based on "modulus two" and hardware duplication techniques.

The "modulus two" monitoring method, in particular the "odd number check," is done byte by byte, which is to say each byte of the data has a check bit that added to the number of units in the byte makes it uneven (odd).

The hardware duplication method is used to monitor the work of the arithmetic-logical unit. By this method each bit of information is processed in direct and inverse form. The results of this duplication are compared for each bit. If the direct and inverse levels coincide this indicates an error.

11,176
CSO: 1863/51

PROBLEMS OF AUTOMATED CONTROL SYSTEMS FOR INDUSTRIAL PROCESSES REVIEWED

Moscow VOPROSY EKONOMIKI in Russian No 9, Sep 81 pp 67-77

[Article by D. Palterovich and M. Gornshteyn: "Automating the Control of Technological Processes in Industry"]

[Excerpt] The national economic efficiency of automated control systems for technological (industrial) processes [ASU-TP's] depends not only on their scientific-technical level, but also on the extent to which they cover existing industrial processes and aggregates, and this figure is still low. In 1979 ASU-TP's had been introduced at slightly more than one percent of all industrial enterprises. Each of these enterprises had, on the average, about two systems. All of the ASU-TP's introduced could be classified as follows: direct control system — 52.7 percent; information-advisor systems — 16.8 percent; information recording systems — 30.5 percent.

We should observe here that the rate of development and extent of application of these systems does not meet national economic needs. Calculations that we have made show that at the end of the 11th Five-Year Plan in the primary sectors of extracting and manufacturing industry, the proportion of industrial facilities equipped with ASU-TP's, although it is growing, is only about 30 percent of the total number of complex industrial facilities prepared for the use of ASU-TP's and at which the use of such systems is economically expedient. These findings illustrate that the national economic need for ASU-TP's is great and cannot be fully met in one five-year plan.

In view of the high efficiency of ASU-TP's, steps should be taken to meet the need for them more fully in the 11th and 12th five-year plans. In our opinion, it would be wise to increase the proportion of expenditures for automation in gross capital investment. In petroleum extraction and nonferrous metallurgy, this proportion is 2-2.5 percent, while in electrical power, ferrous metallurgy, and the gas industry it is 4-5 percent, in the building materials industry — 5.5 percent, in the chemical industry — 8 percent, and in petroleum refining — 7 percent.

For a long time work to build and employ ASU-TP's was carried on in the context of rapid development of ASUP's [automated production control systems]; this could not help being reflected in the overall distribution of capital, and apparently was not always justified. The proportion of investment for ASU-TP's in total capital investment for building automated control systems of all types, despite significant growth in the 10th Five-Year Plan compared to the Eighth, was still insignificant and in the last 10 years has been only 12 percent while investment for ASUP's was 68 percent. This distribution of resources led to a situation where at the beginning of 1980 there were 4,370 automated control systems: 2,460 ASUP's and 1,649 ASU-TP's. At the present time there is less than 0.7 of an ASU-TP for each operating ASUP. But for efficient integrated automated control of industrial and economic-organizational processes at contemporary industrial enterprises there should be from a few to several dozen ASU-TP's for each ASUP. Only in this case can the automated control system for technology become the foundation of an automated production control system (ASUP).

At the same time, because building and operating ASU-TP's requires significant production expenditures,* the paramount challenge is to make optimal use of production resources for this purpose and to determine the rational order and scope of work to set up and employ ASU-TP's.

Analysis of ASU-TP's shows the high economic efficiency of this form of technical progress. One-time expenditures for the development and introduction of ASU-TP's are repaid on the average in 1-3 years, chiefly by reducing use of material and energy resources (by 2-5 percent), increasing the production of output (by 2-8 percent), improving the quality of output, and other factors. The norm of efficiency of capital investment for setting up ASU-TP's was raised slightly for the 11th Five-Year Plan.

In the last three five-year plans several billion rubles have been invested in development of ASU-TP's. Hundreds of scientific research, planning-design, and technological organizations and enterprises and thousands of engineering-technical employees of various ministries and departments are engaged in designing and introducing them. But the level achieved and the results of work to build and introduce ASU-TP's still do not fully meet national economic requirements.

In the first place, as already noted the scale of capital investment in automation is inadequate to supply automated systems to all large industrial aggregates. In the second place, the choice of objects of automation, and consequently the distribution of capital investment among these objects, does not always meet the criterion of maximum national economic impact. In the third place, information and advisor systems still predominate in the structure of existing ASU-TP's. The proportion of control systems is rising slowly, and their functional capabilities are often limited and do not provide, on the one hand, for integration of the ASU-TP and ASUP into a single system for control of technology and production or, on the other hand, sufficient flexibility and adaptability of the industrial facility to change in work conditions. In the

---

* The average cost of setting up one ASU-TP with a computer was about 1.3 million rubles, with an average labor intensity of 90 worker-days.

fourth place, the quality and reliability of the functioning of ACU-TP's often declines because of disproportions in the development of their scientific-technical base and inadequate matching of the computer equipment employed, especially peripheral units and software, to technical and economic requirements. In the fifth place, many ASU-TP's are built as unique developments for one-time application, so the level of unification and standardization of structural, algorithmic, and technical concepts is low and they do not circulate. In the sixth place, the organization of the process of developing ASU-TP's is not always oriented to rapid introduction and fitting the character of the object of automation. In the seventh place, the level of use of ASU-TP's in terms of time and their functional capabilities is inadequate in many cases, which greatly reduces their economic efficiency.

One of the most important problems that must be solved in the area of raising the efficiency of automatation of technological processes in industry is developing the scientific-technical base of automation, improving planning, and straightening out the organizational forms of work related to the development and application of ASU-TP's. New scientific principles and engineering methods of building ASU-TP's must be employed to develop and introduce highly efficient automated technology. The specific ways to accomplish this are determined by both the needs of industry and the capabilities of current automated control equipment.

Contemporary forms of automating control, with their high requirements of industrial equipment, themselves become a powerful factor in the transformation of equipment and technology on the basis of the advances of the current scientific-technical revolution. Automation of control creates conditions for increasing the unit capacity of aggregates and the continuity of their operation, makes it possible to raise temperature, pressure, and other parameters of processes, insures operator safety under conditions of aggressive environments, exposure to radiation, and the like. In this way, the efficiency of ASU-TP's has a clearly marked socioeconomic character. In terms of functional capabilities the most modern domestic ASU-TP's today are as good as the best foreign models and can in certain cases match them for reliability, flexibility, and diversity of technical resources employed. To raise the scientific-technical level of ASU-TP's it is essential to increase significantly the volume of diagnostic, optimization, and control functions in them and to make broader use of progressive scientific methods of control — direct digital control, optimal control by adaptive models, and others (at thermal power units, for example, in 10 years the number of parameters measured has increased six times and the number controlled has risen four times).

Raising the scientific-technical level of ASU-TP's significantly broadens their functional and technical capabilities. For example, the ASU-TP's that were developed earlier for sulfuric acid production facilities were information-dispatcher systems and did not accomplish the tasks of operational control of production. The Kupol ASU-TP that has now been set up for the Gomel' Chemical Plant envisions direct digital control of sulfuric acid production. The basic distinction of this system is its transition from automatic monitoring and analysis of a number of variables to automated regulation of a set of interrelated parameters based on the use of mathematical models and the techniques of combined control, to automatic selection of the best technological regimes and coordination of the work of the entire line. Deviations in the regulated parameters from assigned values were cut by three-fourths to four-fifths with introduction of the Kupol ASU-TP, which made it possible to reduce losses of expensive raw material by one-third to one-half.

The growing complexity of ASU-TP's requires an enlargement of the memory of computer machines, broadening of their links with the object, development of means for operator "communication" with the system, and the like. One of the major scientific-technical advances in building ASU-TP's in the 10th Five-Year Plan was the use of third-generation control computer complexes in most cases. With respect to volume of output and technical characteristics (with the exception of reliability) they meet the requirements of ASU-TP's. But the structure of production of computer equipment still does not meet the national economic need for it. Few specialized computers are available, while general-purpose computers are too expensive and are not always adapted to the requirements of the control objects. Growth in the production of control computer complexes is far outstripping growth in the production of supplementary units: internal memory units, units for communication with objects, and peripheral devices. The structure of production of computer resources should be modified to significantly increase the proportion of peripheral units and units for communication with the object. Industry should also produce a broader assortment of peripheral units (flexible disc stores, specialized production engineer-operator terminals), specialized sensors, actuating mechanisms, and certain other types of automation equipment that are produced in small series.

About one-third of the ASU-TP need for specialized instruments and automation equipment (they constitute 20-25 percent of the total production of technical means for ASU-TP's) to monitor the main parameters of the industrial process is being met. Growth in this need demands the development of essential design subdivisions and experimental facilities oriented to devising not only individual instruments, but entire sets of equipment for ASU-TP's. The report by N. A. Tikhonov at the 26th Congress of the CPSU notes that the production of miniature electronic control machines as a constituent part of the basic industrial equipment, instruments, and various control and monitoring systems and devices is expanding significantly in the 11th Five-Year Plan. This will step up the development of the technical base of the ASU-TP's significantly.

The low reliability of the computer equipment and instruments being produced by industry is a serious obstacle to raising the efficiency of ASU-TP's. This makes it impossible to implement a number of technical concepts (for example, direct digital control) efficiently in industry and leads to parallel use of several computer devices and duplication of the system by manual control. As a result, one-time and ongoing expenditures of the system rise considerably and its economic efficiency declines. A further increase in the efficiency of ASU-TP's demands an improvement in reliability, particularly in the electronic part of the computer, raising the trouble-free period of work to 10,000 hours.

The efficiency of ASU-TP's depends significantly on improving the organization of work to set them up. A specific feature of this work is that it cannot be done apart from the particular characteristics of the machinery and technology of automated production. Technical policy in the field of automation of the control of particular industrial processes and aggregates (production facilities) must therefore be viewed as a part of the technical development or re-equipping of the corresponding production subsector. It must be coordianted with technical policy in the field of development of technology and designing of the primary production equipment.

Expanding the scope of work to set up ASU-TP's demands a transition to more productive, "industrial" methods of developing, implementing, and distributing them to insure an improvement in quality and a reduction in the labor-intensiveness and time of work to set up ASU-TP's. The most important steps to raise the efficiency of setting up ASU-TP's are the following: standardization and unification of design concepts; automation of their design; experimental field testing of pilot models of ASU-TP's.

The rate and scale of development of automation of control over industrial processes is largely determined by the interrelations of the sectors that perform scientific, design, and installation-adjustment work in setting up ASU-TP's, and the manufacturers of the industrial equipment and technical means of control. Comprehensive organization of work to automate the control of industrial processes should be based on uniform intrasectorial long-term programs of scientific research, design , production of technical means, installation, adjustment, preparation of the control object, insuring efficient functioning and development of ASU-TP's, and circulating them.

Organizational forms of work to automate industrial processes have been taking shape for a long time under conditions of the multisectorial structure of industry. Because numerous ministries bring together what are essentially several sectors that differ not only by the nature of their output but also by specific features of industrial processes, work toward automation has been spread out at subdivisions of many different organizations. Even within a particular sector numerous organizations subordinate to different departments work on solving the same technical problems of automating production. All this obstructs the continuity of the process of development and introduction of ASU-TP's.

Certain specialized institutes develop ASU-TP's that are suitable for one-time use at a particular site, which retards the formulation of standard ASU-TP's for widespread introduction. Work sometimes goes forward in parallel at several enterprises, and the result is that different ASU-TP's are developed for identical enterprises. But experience shows that the total length of working time to set up an ASU-TP using standard concepts is 30-40 percent less than where there is individual development, and costs are 20-30 percent lower.

The main weaknesses of the organization of work today to set up ASU-TP's result from the following factors, among others: the absence of uniform leadership and technical policy in the field of planning, coordinating, and monitoring this work; the lack of specialized organizations responsible for technical policy and the organization of work to automate certain types of industrial processes; duplication in solving the same scientific and technical problems; and, the unsatisfactory state of experimental production facilities.

To improve the efficiency of work on ASU-TP's their organizational forms must be coordinated with the content and scope of work on automation. The time has come to develop a long-term program of scientific research and planning-design work on automating the control of industrial processes, including the manufacture of pilot models of automated control systems for complex industrial processes, development of mathematical models of control objects, development of standard algorithms and programs for ASU-TP's, and so on. The sectors should have long-term plans for setting up ASU-TP's, and the resource requirements established by these plans must be taken into account in five-year and annual plans.

Solving the problems of automating the control of industrial processes, which requires substantial labor, material, and financial resources, can only be done by large organizations. The most rational form of these organizations is the science-production association. But there are still very few of these associations. To avoid duplication in work to develop and introduce ASU-TP's in various sectors, a number of steps should be taken to determine the specific specializations of organizations in automating similar industrial processes and main designers should be appointed for groups of processes. These organizations must be responsible to the customer for performance of the full range of work related to ASU-TP's.

In our opinion, work to set up pilot automated control systems for complex industrial processes should be concentrated at organizations of the Ministry of Instrument Making, Automation Equipment, and Control Systems. After testing the pilot systems should be turned over to sectorial organizations for distribution. This is a useful system of work because, in addition to the fact that this ministry has strong scientific, planning, and design organizations involved with ASU-TP's, of the common features of methods of studying industrial processes as control objects, the possibility of standardizing the information being used, and the general industrial character of monitoring and regulation equipment and computer technology.

The complexity of the equipment used in ASU-TP's necessitates a large number of highly qualified service personnel. Qualified specialists are needed for the software of the control computers and also to repair them. Therefore, it is important to improve the organization of work to service the control computer complexes that are employed in ASU-TP's by establishing a procedure for centralized servicing by the manufacturing ministry. This will make it possible to improve the quality of service and reduce its cost.

Among the factors that are slowing down the introduction of automation at industrial enterprises and increasing the time of work we may also include unsatisfactory material-technical supply. In various sectors the planning of material-technical supply for work on automation is often done apart from the time and scope of the work, and resources planned for automation are sometimes spent for other purposes. Therefore, we feel that special-purpose funds should be established for material-technical supply of automation work and they should be coordinated with the volume of capital investment in automation.

Poor material-technical support for research and development and automation has a negative effect on work to set up ASU-TP's. In addition, the organizations which are doing this work must submit orders (requests) for technical means and materials one year before the beginning of the work mentioned above; in some cases this leads to failure to supply essential materials and articles, while in other cases above-norm stocks of scarce materials are formed, working capital is frozen, and so on. We believe that it is necessary to establish an organizational solution so that orders for technical means and materials for conducting scientific research and experimental design work are filled within three months.

The efficiency of setting up ASU-TP's is often lowered by incomplete supply of equipment. Individual assemblies, mechanisms, and instruments in unitary industrial aggregates equipped with automation means must be installed by primitive methods in the local areas. Both primary and auxiliary industrial equipment should be designed with built-in sensors and actuating mechanisms or with assigned places where they are built in and then delivered by the manufacturing enterprises together with automation instruments and equipment. At many enterprises, especially sectors where equipment operates under difficult conditions (chemistry, petrochemistry, metallurgy, the fuel industry, and others), ASU-TP's are used inefficiently because of a lack of spare parts. We must increase spare parts production considerably (to 25 percent of the volume of production of automation equipment) and create reserves of such parts. It would be wise to plan to meet the spare parts needs of operating ASU-TP's on an equal basis with supply of equipment to start-up projects.

In a number of cases the setting up of automated technology is held back by failure to insure that projects planned for control by traditional methods and means are technically and organizationally prepared for automation. For many years automation was thought to be an independent stage of activity toward which one should move on the basis of using already-existing industrial equipment. The typical negative aspect of the process of automation at existing enterprises was (and still is today in many cases) "building on" automation to existing equipment which is ill-suited for work under conditions of automation of control. Often even the new equipment proves unsuited for automatic regulation or for the volume of control functions included in the ASU-TP.

As work on automation of control progressed it became increasingly clear, on the one hand, that it was not wise to switch certain aggregates and processes (obsolete or unpromising ones) to automated control and, on the other hand, that industrial aggregates had to be rebuilt when transferred to automated control. Progress in the field of automating the control of industrial processes and a guarantee of its economic efficiency lie not in automating aggregates designated for manual control, but rather in setting up automated aggregates and industrial complexes that are oriented to automated control from the beginning of their development.

ASU-TP's should be developed parallel with the creation of new technology and new industrial equipment. To insure that ASU-TP's are launched in operation together with the launching of production facilities it is necessary to carefully plan the times at which ASU-TP's are set up at construction projects and sites being rebuilt. This procedure should be codified by working out norms and requirements for industrial equipment designated for work in an automated mode.

Widespread and efficient development of ASU-TP's is possible only if there is allout standardization of technological concepts, diagrams, and complexes. When technological diagrams are not standardized the result is greater complexity of concepts during automation and difficulties in distributing the system. In addition, raising the level of technological standardization makes it possible to reduce the number of design stages and cut the realization time and cost of the system. Broadening the sphere of rational application of ASU-TP's is also linked to growth in mass production, the need to insure its smooth rhythm, and the necessity of increasing the stability of the composition of raw materials and basic industrial parameters. Many steps toward automation (for example in ferrous metallurgy) are taken in sections that have significant equipment downtime, violate production schedules, and suffer constant fluctuations in the content of raw materials.

In order to carry out the measures to accelerate the launching of production capacities and facilities and raise the efficiency of capital investment as outlined in the decree of the CPSU Central Committee and USSR Council of Ministers on improving the economic mechanism, it is essential to work out a sound approach to selecting the sites for which ASU-TP's are planned. The determining characteristics when selecting an industrial site to be equipped with a computer-based ASU-TP and when establishing the class of the particular control system should, in our opinion, be the following: the future promise of the industrial process, its degree of refinement and level of standardization from the standpoints of nature and structure; the dimensions of the unit capacity of the installation; the value and scarcity of the output produced; the complexity of the process from the technological and operating standpoints; the frequency, level, and economic consequences of disturbances that affect the industrial process; the frequency of changes in the production situation; the level of supply of monitoring and measuring instruments; and, the nature of internal (production conditions) and external (marketing conditions) constraints on production of output.

Analysis of the technical-economic prerequisites of automation is very important when selecting a control object. Underestimation of the role of this analysis may lead to one of the significant shortcomings that have been experienced in planning the development and use of ASU-TP's in industry. Often the decision to set up an ASU-TP is inadequately substantiated relative to the particular site. This relates above all to determining the technological potential of the site and the potential of control; identifying, evaluating, and analyzing production losses related to the level of control; and, evaluating the preparedness of the site for introduction of an ASU-TP.

It is difficult to overcome this problem because the customer, when selecting the object and determining the level of its automation, is not at all responsible for the economic and technical wisdom of the decision being made. It appears that there should be a fundamental re-evaluation of the role and content of the stage of "technical-economic substantiation of setting up an ASU-TP," in which the goals of setting up the ASU-TP are formulated, its structure is laid out, and interrelationships with the control system are established. It is in this stage, in fact, that the foundations should be laid for the use qualities of automated control systems that make it possible to guarantee the client that it will have at least normative economic efficiency.

The number and quality of ASU-TP's are determined above all by national economic need for them and the resources of the scientific-technical base. It is therefore necessary to analyze the national economic need for ASU-TP's in order to improve the planning and organization of work to set them up. This work is now done from one case to the next and does not have any kind of scientific-methodological substantiation, which leads to great divergences in estimations of this need. The estimation of sectorial need for ASU-TP's made in 1977 for the appropriate period revealed that the estimate was several times less than the one established by earlier estimates done in 1973. The primary reason for the discrepancies was that during the period of compiling the 1973 estimates the sectors did not have an adequately clear idea of the essential nature of ASU-TP's and their jobs and place in the overall system of production control. When indicating their need for ASU-TP's, the sectors in effect estimated the need for other forms of automation, including local automation. needs are overstated or understated as a result of an incorrect estimation of the technical feasibility and economic wisdom of introducing ASU-TP's for specific industrial facilities.

The ASU-TP's now being developed and introduced are very diverse both with respect to scale and nature of problems solved and the functional-algorithmic structure corresponding to them and ways and means of solving them. Therefore, in order to estimate the national economic need for ASU-TP's, to plan for setting them up and provide production resources for this work, and to monitor performance of the assignments of national economic plans the full set of ASU-TP's should be broken down into subsets (groups) characterized by definite features. The most significant of these features are the following: (a) nature of the occurrence of the controlled industrial processing time — degree of continuity of arrival of raw material, relative length of particular operations; (b) degree of functional development — complexity of information and control functions realized; (c) information output — the number of technological variables measured or monitored by the ASU-TP.

In our opinion, to improve the planning and organization of work to set up ASU-TP's and to distribute production resources for automation of control it would be advisable to work out a system of norms. This system should include norms for specific capital investment to set up ASU-TP's of different classes, labor expenditures for different stages of setting up and using them, service lives, and expenditure of operating materials.

On a national scale it is necessary to conduct factory certification (by "passports") of the most important types of automatable industrial processes according to their basic technical-economic characteristics as control objects. This will make it possible to identify the primary kinds of control objects, group objects by types, establish the characteristic features of automatable processes, aggregates, and facilities as control objects, and more closely coordinate the creation of new technological processes and aggregates with automation of their control, thus insuring maximum economic efficiency.

In the current phase of development of ASU-TP's, during the period of their growing distribution, it is necessary to take a large number of economic-organizational measures related not only to the development of systems (their organizational structure and preparation of the objects), but also to production control under conditions of existing ASU-TP's, further development of functioning systems, and raising the efficiency of automated control. Many of these economic-organizational questions are either being decided in an unsatisfactory manner or simply not considered at all. Thus, there is no comprehensive development underway on questions of improving the control system and methods of planning, monitoring, recording, and analyzing the production-economic activity of the enterprise under ASU-TP conditions. Proper attention is not being given to improving control processes under ASU-TP conditions, or to economic stimulation of ASU-TP developers and operations personnel (giving them incentive to improve the efficiency of control systems).

We do not today have a clearcut system of economic stimuli for the development, operation, and continuous refinement of ASU-TP's. In most industrial sectors the wages of automation equipment service personnel are lower than those of primary production workers. We must eliminate this inequality and increase the icnentive for production and management personnel to use ASU-TP's efficiently.

Precise organizational forms for managing the development, introduction, and elaboration of ASU-TP's have not been established at the enterprises. There often is no specialized subdivision in charge of setting up and elaborating the ASU-TP. The number of ASU-TP service personnel at most industrial control objects is inadequate. The monitoring-measuring instrument and automation shops are inappropriate for the jobs of operating ASU-TP's.

There is no statute on ASU-TP services at enterprises which would establish appropriate legal norms and rigidly define the accountability of the client and developer during the setting up and use of an ASU-TP. The client is not accountable for formulation of the specific technical-economic goals of setting up the ASU-TP, for the economic and technical wisdom of the decision he makes, for timely preparation of service personnel, for general preparation of the object for work under ASU-TP conditions, and for insuring that the system is used efficiently. The functions and responsibility of the client during development and use of the ASU-TP should, in our opinion, be clearly established in a special statute ratified by the USSR State Committee for Science and Technology.

Along with other paths to improving ASU-TP's that have been considered, solving the problems of economic-organizational support of these systems will make it possible to improve the efficiency of their setting up and use in industry.

11,176
CSO: 1863/46

HYBRID COMPUTERS

HYBRID COMPUTING MACHINES AND SYSTEMS: LOCAL AUTOMATED CONTROL SYSTEMS AND COMPUTER DEVICES

Kiev GIBRIDNYYE VYCHISLITEL'NYYE MASHINY I KOMPLEKSY: LOKAL'NYYE ASU I USTROYSTVA VYCHISLITEL'NOY TEKHNIKI in Russian No 4, 1981 (signed to press 6 May 81) pp 2, 117-121

[Annotation and abstracts of articles in collection "Hybrid Computing Machines and Systems: Local Automated Control Systems and Computer Devices", edited by G.Ye. Pukhov (editor-in-chief), et al., Izdatel'stvo "Naukova dumka", 1000 copies, 121 pages]

[Text] Annotation

This collection presents the results of scientific research on the theory, methods and algorithms for hybrid computation, on the development of computer and peripheral devices, and on analyzing the accuracy, reliability and diagnostic methods of hybrid systems. Some problems in developing hybrid local automated control systems are considered.

The collection is intended for local automated control system and computer hardware developers, as well as students, graduate students and scientific workers specializing in the area of hybrid computer technology.

UDC 536.629+681.34

APPROXIMATION OF DYNAMIC CHARACTERISTICS OF OBJECT WITH DISTRIBUTED PARAMETERS USING RC-NETWORK MODEL

[Abstract of article by G.V. Biryukova and G.V. Yevstratov]

[Text] It is shown possible to obtain a homomorphic mathematical model of an object with distributed parameters (DP-object) using a hybrid computer system (HCS) containing a dynamic analog model of the thermal object in question in the RC-network. An algorithm is given for approximating the dynamic characteristics of the DP-object using an ordinary second-order differential equation with a delayed argument. Four bibliographic references.

UDC 681.33

DIGITAL MODELING OF DISCRETE-ANALOG NETWORK PROCESSOR

[Abstract of article by G.N. Azarov and V.Ye. Prokof'yev]

[Text] Problems of investigating the accuracy of RC-models on pulse-time adjustable conductances using digital modeling methods are examined. It is shown that there is no propagation of methodological error in the discrete-analog network model from node to node, and the methodical error is obtained as a function of the model parameters. Two illustrations, one table, one bibliographic reference.

UDC 681.3

HYBRID DEVICE FOR DEFINING CORRELATION FUNCTIONS OF NORMALLY DISTRIBUTED STATIONARY RANDOM PROCESSES

[Abstract of article by V.F. Kornilovskiy]

[Text] A device is described for defining the correlation functions of normally distributed stationary random processes. The device consists of a digital (sign) correlator and a digital-analog converter. The device uses series 155 integrated circuits. Four illustrations, three bibliographic references.

UDC 518:517.944/947

MODELING OF PROCESSES OF INTRA-RESONATOR GENERATION OF SECOND OPTICAL HARMONIC CONSIDERING TRANSVERSE HETEROGENEITY OF RADIATION

[Abstract of article by A.A. Glushchenko, B.P. Dovgiy, V.V. Obukhovskiy and V.L. Strizhevskiy]

[Text] Equations modeling the subject generation are investigated numerically with the combined influence of nonlinear, diffraction and dissipative losses for various gains. The optimal linearity which provides the greatest intensity of the second harmonic is found. Four illustrations, 11 bibliographic references.

UDC 681.3.06:681.34

ALGORITHM FOR ANALYZING PATCHING SCHEME OF OPERATIONAL MODULES USED TO SIMULATE AUTOMATIC PROGRAMMING SYSTEMS FOR ANALOG COMPUTERS

[Abstract of article by A.N. Klimenko]

[Text] An algorithm is described for analyzing the patching schemes of operational modules in the analog section of a hybrid computer section which can be used to select the optimal version from the viewpoint of reducing the number of modules used in the analyzed circuit. The structure of the algorithm and an example of its use are examined, and the results are discussed. Two illustrations, two bibliographic references.

UDC 62-503.3

SOME ALGORITHMS FOR IMPLEMENTING CONVOLUTION OPERATOR AND THEIR APPLICATION

[Abstract of article by A.F. Verlan' and B.B. Abdusatarov]

[Text] Various questions involved in the numerical realization of linear integral operators and Voltaire equations of the second sort with arbitrary and partionable kernel are considered. A series of functional diagrams of special-purpose computers used to implement integral operators and solve integral equations are also presented. Seven illustrations, three tables, four bibliographic references.

UDC 681.33

DEVICE FOR AUTOMATED INPUT OF PARAMETERS OF DISCRETE-ANALOG NETWORK MODEL

[Abstract of article by V.V. Garmash]

[Text] A device for inputting network model parameters and interfacing with the memory circuit of the node element is examined which can be used to fully automate the process of inputting RC-network parameters. The device can be used as the basis for creating a "RC-network -- digital computer" hybrid system.

UDC 681.3.056

WIDEBAND CODE-TIME INTERVAL CONVERTER

[Abstract of article by A.N. Bazhenov and Yu.I. Gerashchenko]

[Text] A converter is examined which uses information from a digital computer or code assignment device to form a time interval in eight ranges (from 10^{-7} to 10^3 sec) in increments of 0.1% of the maximum limit of the range. The converter is implemented using series 155 integrated circuits with minimum hardware. One table, one illustration, four bibliographic references.

UDC 621.314

ANALYSIS OF INFLUENCE OF AVERAGERS ON CODE-TIME INTERVAL CONVERSION ERROR

[Abstract of article by M.G. Rokhman]

[Text] This article analyzes the influence of buffer averagers connected to the input of an adjustable frequency divider on variation of the error of a code-time interval converter. The most characteristic combinations of binary sub-harmonic components of the output frequency of the adjustable frequency divider are examined. One illustration, four bibliographic references.

UDC 621.317.08

INVESTIGATION OF ROTATION SPEED-TO-PULSE FREQUENCY FEEDBACK CONVERTER

[Abstract of article by V.I. Dotsenko and B.A. Furman]

[Text] This article demonstrates the advantage of modulating the U-factor of the sensing element of a feedback converter prior to modulating the feedback coefficient. A method for improving the converter resolution is examined, and recommendations are given for the choice of frequency and operating clearance. Three illustrations, two bibliographic references.

UDC 681.34

PROCEDURE FOR CONFIGURING PATCH PANEL WITH MATRIX SWITCHES FOR HYBRID COMPUTER SYSTEMS

[Abstract of article by V.N. Gugnin]

[Text] This article considers configuration of the patch panel and logical description of the modules in a device for controlling an automatic patch panel using structural matrices of acceptable connections between decision modules in the analog section of a hybrid computer system. The application of the configuration of the patch panel of two-stage switching circuit is demonstrated using an example. Eight illustrations, two bibliographic references.

UDC 681.33

NODE ELEMENT OF DISCRETE-ANALOG NETWORK PROCESSOR

[Abstract of article by V.M. Andriyevskiy]

[Text] Questions involved in constructing a node element of a discrete-analog network processor based on pulse-time adjustable conductances are considered. The advantages and disadvantages of these node elements are pointed out, and ways of improving their accuracy and expanding their functional capabilities are examined. The functional diagram of a node element with memory devices which assumes the use of a digital computer to automate parameter setting is proposed. Three illustrations, six bibliographic references.

UDC 681.333(088.8)

SIMULATION OF DOMAIN-BOUNDARY FORMATION MODULES USING ALGEBRAIC LOGIC

[Abstract of article by K.A. Babordin]

[Text] The application of the apparatus of algebraic logic is demonstrated for formalizing the development stage of the circuits used in the domain-boundary formation modules which make up probabilistic hybrid computer systems which are oriented towards solving boundary problems. Five illustrations, three bibliographic references.

UDC 681.326.74

METHODS FOR ORGANIZING DIAGNOSIS OF SPECIAL-PURPOSE PROCESSORS AND DEVICES

[Abstract of article by R.S. Khalatyan]

[Text] This article examines questions involved in organizing microdiagnosis of special-purpose processors (SP) and devices using diagnostic instructions represented by special microprograms. A generalized SP structure is presented, along with the sequence used by the special processor to execute the diagnostics. Besides a two-level (program and microprogram) organization of SP diagnostics, another version is proposed which combines analysis of the response of the SP with a procedure for diagnosing it at the microprogram level. The possibility of diagnosing peripheral devices without operator intervention is indicated. Two illustrations, 16 bibliographic references.

UDC 681.326.7

FULLY SELF-VERIFYING TWO-LEVEL CONTROL CIRCUITS FOR m OF 2m+1 AND m+1 OF 2m+1 CODES

[Abstract of article by V.V. Neshveyev]

[Text] A simple method is proposed for partitioning a set of code words of an m of 2m+1 and m+1 of 2m+1 codes into two non-intersecting subsets which allows for the structure of these codes.

It is fairly easy to use the partitioning obtained to construct fully self-verifying control circuits for m of 2m+1 and m+1 of 2m+1 codes. One illustration, five bibliographic references.

UDC 621.382

AUTOMATED MONITORING SYSTEM FOR DISCRETE DEVICES

[Abstract of article by V.V. Antosik, P.M. Demochko and R.I. Krutykh]

[Text] This article examines an automated control system controlled by an "Elektronika 100/16I" computer which can be used for functional control of discrete devices with up to 512 outputs. The functioning accuracy of the control system with single-output and two-output control circuits is estimated. One illustration, four bibliographic references.

UDC 658.5;681.3;62--501.72

METHOD FOR DESIGNING DIGITAL SYSTEMS FOR REGULATING TECHNOLOGICAL PARAMETERS WITH SELF-TESTING

[Abstract of article by T.G. Mashchenko, O.I. Potepukh and T.A. Yakovenko]

[Text] This article examines the possibility of self-testing during functioning of digital systems which regulate technological parameters. Duplication testing is used to organize system self-testing. A digital system for regulating technological parameters is used as an example for examining the methodology of designing self-testing systems. Three illustrations, four bibliographic references.

UDC 681.324

ESTIMATING EFFICIENCY OF SPECIAL-PURPOSE COMPUTERS

[Abstract of article by Sh.Sh. Agzamov]

[Text] Criteria are proposed for estimating the efficiency of special-purpose computers which allow for the specific characteristics of the computers, the problems to be solved, and the specific requirements of practical utilization of the devices. Allowance is made for limitations on the parameters and the influence exerted by each parameter on the operation of the device. The efficiency of special-purpose computers is estimated with respect to a single abstractly simulated standard structure which has optimal parameters and solves one specific problem. One illustration, seven bibliographic references.

UDC 681.142.65

COMPARATIVE ESTIMATION OF ACCURACY AND SPEED OF COMPUTERS IN SOLVING STANDARD PROBLEMS -- A REAL-TIME RING TEST

[Abstract of article by Sh.Sh. Agzamov]

[Text] This article considers problems involved in comparative estimation of the accuracy and speed of the EMU-10 analog computer and of digital computers in solving the ring test problem (standard problem) in real time.

A comparative estimate of the accuracy and speed of an M-7000 digital controller and a digit-analog computer system is made for solving a standard problem using the first-, second- and fourth-order Runge-Kutt method. One illustration, five tables, six bibliographic references.

UDC 621.391.175

ANALYSIS OF CONTROL SYSTEM DATA PROCESSING DEVICES

[Abstract of article by G.F. Krivulya and A.A. Ushakov]

[Text] This article examines the organizational principles of data processing devices in control systems with a "pass-fail" classification of the object. The estimate and results of computer modeling of different methods of organizing control systems are presented. Two illustrations.

UDC 621.9.06--529: 681.323

EFFICIENCY OF UTILIZING INTERPOLATION ALGORITHMS IN NUMERICAL PROGRAM CONTROL SYSTEMS

[Abstract of article by V.D. Baykov and S.N. Vashkevich]

[Text] The use of additional microcircuits to increase the capacity of microprocessor devices is proposed for numerical program machine tool control systems. Quantitative estimates are given for interpolation algorithms with and without additional microcircuits, and the effect of their use is evaluated. A criterion is cited which demonstrates the possiblity of increasing microprocessor capacity by using additional microcircuits. Two tables, six bibliographic references.

UDC 681.325.65

AUTOMATION OF ADJUSTMENT OF ELECTROTECHNICAL LSI TOPOLOGY TESTING MODULE IN SAPR USING SYMBOLIC METHOD

[Abstract of article by V.P. Rubtsov and A.M. Abbsov]

[Text] A method is examined for automating electrotechnical testing of the symbolic plan and then the topological drawing of large-scale integrated circuits. The task is reduced to the solution of a system of logical equations. The method has sufficient technological stability and guarantees complete testing. Three illustrations.

UDC 681.34

COMPUTER CAPACITY IN EXECUTING FUNCTIONAL DATA TRANSFORMATIONS

[Abstract of article by N.I. Korsunov]

[Text] The capacity of digital, analog and hybrid computers in computing functions is examined on the basis of the input data transformation rate.

Relationships are obtained which can be used to determine the capacity of computers and the necessary memory load. The efficiency of using various types of computers for function computation is analyzed, and an example is presented and used as a basis for obtaining numerical results. Five bibliographic references.

UDC 621.3.019.3

SPECIAL STAGE IN OPERATION OF DIAGNOSTIC AUTOMATIC PROCESS CONTROL SYSTEM ACCOMPANIED BY MANIFESTATION OF DEFECT

[Abstract of article by V.P. Shargovskiy and S.A. Strel'tsov]

[Text] A technical diagnosis stage is isolated -- from the instant the occurrence of a defect is registered to the instant the tested object and the technical devices associated with the diagnostic automatic process control system are disconnected. A justification is provided for the importance of accelerating preliminary locations of a defect of the tested object as a whole, or the automatic process control system as a whole. Two illustrations, three bibliographic references.

UDC 681.33

SYNTHESIS OF OPTIMAL DIGITAL-ANALOG REGULATOR FOR CONTROLLING THERMAL OBJECT

[Abstract of article by G.V. Yevstratov]

[Text] This article provides a brief review of different approaches to synthesizing optimal control of thermal objects, noting their shortcomings. Based on the example of synthesizing a digital-analog regulator which is optimal in terms of speed it is shown that this problem can be solved efficiently on a hybrid computer system containing a dynamic model-analog of the thermal object in the RC-network circuit. Four illustrations, eight bibliographic references.

UDC 62-83

TECHNICAL LINEARIZATION OF CONTOURS OF DIRECT DIGITAL REGULATION

[Abstract of article by A.I. Ovcharenko]

[Text] An original device for technical linearization of direct digital regulation systems is investigated; speed and error characteristics are obtained. Three illustrations, two bibliographic references.

UDC 62-50-503

PARAMETRIC IDENTIFICATION OF HIGH SPEED AUTOMATIC REGULATION SYSTEM CIRCUITS WITHIN CLASS OF LINEAR IMPULSE MODELS

[Abstract of article by A.I. Ovcharenko]

[Text] An approach is proposed for synthesizing digital automatic regulation systems based on a priori knowledge of the parameters of the linear analog system-model. Criteria are proposed for the correspondence of the impulse and continuous models and parametric identification of the impulse linear model is done. The condition for suppressing periodic modes caused by level quantization in the analog-digital converter is obtained. Two illustrations, six bibliographic references.

UDC 681.3.02

ARRANGEMENT OF COMPONENTS ON PRINTED CIRCUIT BOARD SUBJECT TO VIBRATION

[Abstract of article by E.N. Rybnikov and A.I. Khyannikyaynen]

[Text] This article examines the problem of optimal arrangement of electronic radio components on a circuit board subject to vibration. A foundation is provided for the optimization criterion -- probability of failure-free operation. The problem is stated in general form, and limitations are formulated. Five bibliographic references.

6900
CSO: 1863/64

UDC 681.3.06:681.34

ALGORITHM FOR ANALYZING PATCHING SCHEME OF OPERATIONAL MODULES USED TO SIMULATE AUTOMATIC PROGRAMMING SYSTEMS FOR ANALOG

Kiev GIBRIDNYYE VYCHISLITEL'NYYE MASHINY I KOMPLEKSY: LOKAL'NYYE ASU I USTROYSTVA VYCHISLITEL'NOY TEKHNIKI in Russian No 4, 1981 (signed to press 6 May 81, manuscript received 10 Jan 80) pp 20-23

[Article by A.N. Klimenko from book "Hybrid Computers and Systems: Local Automatic Control Systems and Computer Devices", edited by G.Ye. Pukhov (editor-in-chief) et al., Izdatel'stvo "Naukova dumka", 1000 copies, 121 pages]

[Text] In designing hybrid computer systems (HCS) with automated analog computer programming systems (AACPS), the problem of arranging the operational units of the analog section in the HCS design units, or modules, must be resolved [1].

The quantitative and qualitative makeup of the HCS modules combined by a common hierarchical switching system, and of the problem set [2], must be established a priori in each specific HCS during the design process. In order to minimize developmental hardware costs without limiting the functional capabilities of the future HCS with AACPS it is necessary to make a preliminary analysis of the operational unit patching schemes synthesized by the AACPS for various mathematical descriptions of the systems of differential equations to be solved. As a result, this analysis should provide material for optimal arrangement of the operational units by modules and to determine the nature, type and number of connections between operational units within and without a module.

This article describes an algorithm for analyzing the patching circuits of operational units in the analog section of an HCS. The structure of the algorithm and an example of its utilization are examined, and the results are discussed.

The criterion for optimal arrangement of operational units in modules is an indicator which makes it possible to concentrate operational units in each module which have the maximum number of interconnections and minimal interconnections with units which are part of other modules.

The unit combination efficiency indicator

$$v = r/(r + s), \tag{1}$$

where r is twice the number of interconnections between the operational units which are proposed to be contained in a single module; s is the number of external connections of these units.

Figure 1 shows the flowchart of this algorithm. The initial data for the program consists of the following files: string array SMT[1:T] -- the module composition by types of operational units; integer array SMK[1:T] -- module composition in terms of number of like operational units; boolean array P[1:N,1:F] -- an ordered file of connections between operational units in the circuit being analyzed; string array MS[1:N] -- an ordered file of the operational unit types which are part of the circuit being analyzed.

Here N is the number of operational units in the circuit; F = N + Q, where Q is the number of nonstructural inputs and outputs in the operational unit patching circuit.

Block PM1 executes the following operations:

-- converts file SMK to an analogous file MKT [1:KRT] by removing those operational unit types from SMK which are not used in the circuit being analyzed;

-- forms file MN [1:KRT], which contains information about the maximum number of different types of operational units in the circuit being analyzed;

-- computes M, which is the maximum attainable number of operational units in the module for a particular circuit;

-- pads files RP and SP [1:N] with values of r_i and s_i computed by file P for each operational unit.

Block EF1 computes the efficiency of combination for two operational units connected directly to one another.

Block EF2 computes the efficiency of combining MTEC = 3, 4, ..., M operational units in a module. The combination of operational units in the circuit which are acceptable from the viewpoint of qualitative and quantitative module composition are examined for each value of MTEC, and file KORD [1:N(M-2)] is used to store the combination with maximum efficiency which contains that operational unit for each of the N operational units. File EF 2 [1:N(M-2)] stores the current maximum efficiency values achieved during the sorting, while file KORD stores the coordinates of the combined operational units.

If this section of the algorithm were to be implemented on the basis of a combination generator with complete trial and error, the time expended would be estimated by the number

$$KC = \sum_{\mathrm{MTEC}=3}^{M} C_N^{\mathrm{MTEC}}. \tag{2}$$

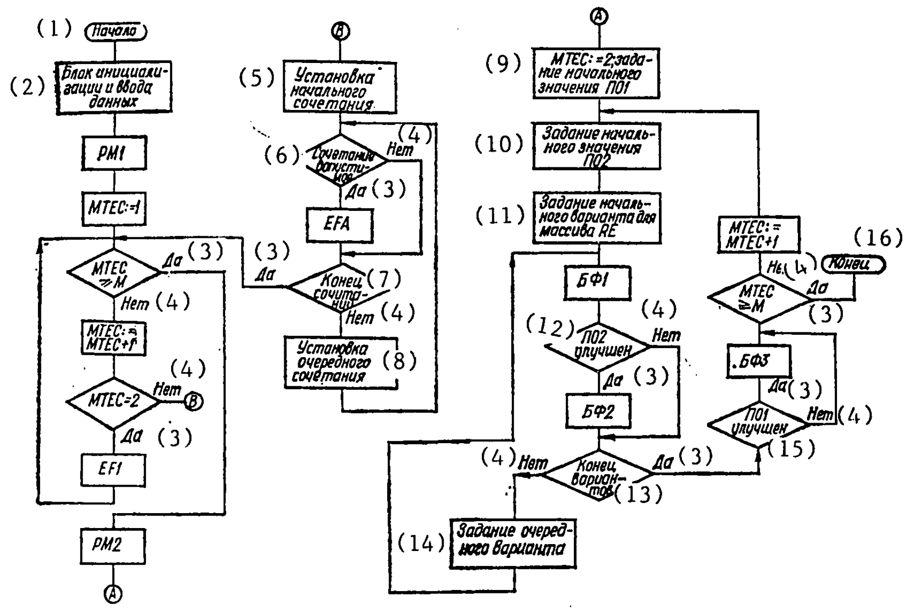


Figure 1.

Key:

1. start
2. initialization and data input block
3. yes
4. no
5. set initial combinations
6. combination acceptable?
7. end combination
8. set next combination
9. MTEC: =2; assign initial value PO1
10. assign initial value of PO2
11. assign initial version for file RE
12. PO2 improved?
13. end of alternate?
14. assign new version
15. PO1 improved?
16. end

The number of trials is reduced by supplementary analysis of the conditions for placing an operational unit in a module allowing for limitations with respect to number and type contained in files MN and MKT.

After eliminating identical combinations from file KORD, block PM2 forms binary file NOM [1:MIKON[M-2]], whose boundaries are stored in file MIKON [1:(M-2)] for each value of MTEC, for each value of MTEC. These are then used as the values of the variable IKON.

The next segment of the flowchart of the algorithm processes file NOM, which contains the set of combinations which can be used as the basis for organizing modules. Some of these combinations form nonintersecting sets, i.e., they contain different operational-unit numbers which are assigned to the operational units during synthesis of the functional diagram. These combinations correspond to maximal utilization of the capabilities of the hardware content of the modules when they contain the operational unit of a specific circuit. The choice of combinations which are nonintersecting sets in the aggregate is not unique. The algorithm selects as optimal the version which corresponds to one of the maximal numbers of nonintersecting sets. This version provides the minimum number of residual combinations at the same time. These occur as the result of eliminating the operational-unit numbers which belong to the nonintersecting sets from the combinations which comprise intersecting sets. The residues, which are subsets, are eliminated. Finally, the number of design modules fitted in a particular circuit will be minimal.

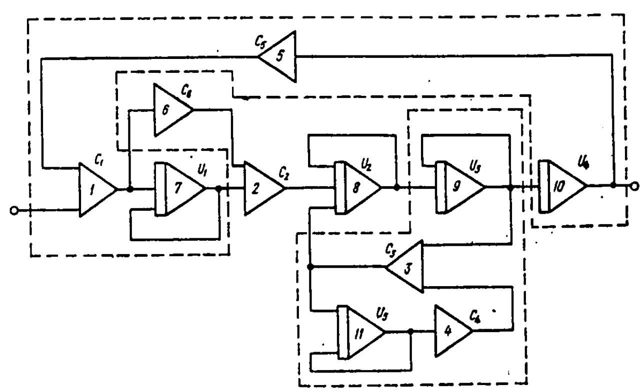


Figure 2.

File RE [1:IKON] is used to store the intermediate results. The final results for each value of MTEC are stored in file REO [1:IKON]. The optimal final result of all MTEC values obtained is selected. This is entered in file REOPT [1:IKON]. These files are formed by blocks BF1, BF2 and BF3. Optimality indicators PO1 and PO2 are used in searching for the optimal version.

The program which implements this algorithm is written in PL/I and includes only 300 operators. The debugging was done on a YeS-1020 computer. Program translation and editing requires 9 minutes. The example presented below was solved in 2 minutes.

The functional diagram of the operational unit patching scheme shown in Figure 2 is used as the example.

Example. Initial data: SMT [1:T] = C, H, Y, U, SMK [1:T] = 2, 2,3,2; N = 11; F = 23; P[1:N,1:F] = 0,0,0,0,1,1,1,0,0,0,0,1,0;0,0,0,0,0,1,1,1,0,0,0,0,0;0,0,0, 1,0,0,0,1,1,0,1,0,0,;0,0,1,0,0,0,0,0,0,0,1,0,0,;1,0,0,0,0,0,0,0,0,1,0,0,0;1,1, 0,0,0,0,0,0,0,0,0,0,0;1,1,0,0,0,0,1,0,0,0,0,0,0;0,1,1,0,0,0,0,1,1,0,0,0,0;0,0, 1,0,0,0,0,1,1,1,0,0,0;0,0,0,0,1,0,0,0,1,0,0,0,1;0,0,1,1;0,0,0,0,0,0,1,0,0.

Here C, H, Y, U are the names of the operational units.

Results:

M = 4; KRT = 2; MKT [1:KRT] = 2,2;
MN [1:KRT] = 6,11;
RP [1:N] = 0,0,0,0,0,0,2,2,2,0,2;
SP [1:N] = 4,3,4,2,2,2,2,3,3,3,2;
MIKON [1:(M-2)] = 5,11;
MTEC = 3;
NOM [1:5] = 1,5,7;2,6,7;3,4,11;3,8,9;5,9,10;
EF2 [1:5] = 0.600; 0.666; 0.800; 0.714; 0.600;
REO [1:5] = 2,6,7;3,4,11;5,9,10;1;8;
MTEC = 4;
NOM [1:6] = 1,2,7,8;2,6,7,8;3,4,8,11;3,5,8,9;3,4,4,9,11;1,5,7,10;
EF2 [1:6] = 0.625; 0.714; 0.800; 0.625; 0.800; 0.615;
REO [1:6] = 3,4,9,11;1,5,7,10; 0; 2,6,8;0;0;
REOPT [1:6] = 3,4,9,11;1,5,7,10;0;2,6,8;0;0.

The dotted lines in Figure 2 set apart the sections of the circuit which the program placed in different design modules in accordance with the data held in file REOPT.

In terms of the number of combinations tried, the following results were obtained for this example. With a complete exhaustive search of (2)

$$KC = C^3_{11} + C^4_{11} = 165 + 330 = 495.$$

Considering the limitations included in the algorithm, this number of combinations was

$$KC_{trunc} = 151 + 199 = 350.$$

The number of combinations for which the efficiency is computed is

$$KC_{accept} = 135 + 150 = 285.$$

It follows from this data that in this example when exhaustive search is used, 210 combinations are unacceptable, while 65 are unacceptable for the truncated search.

We note in conclusion that repeated application of this program to different functional diagrams of operational unit patching makes it possible to work out an optimal composition of modules for a given class of problems solved by an HCS, and to obtain statistical data for equipment loading and configuring automatic HCS patching systems.

BIBLIOGRAPHY

1. Hannauer, G. Automatic patching for analog and hybrid computers. Simulation, 1969, 12, N 5, p. 219-232.

2. Kalashnikov, V.I., Klimenko, A.N. "Approach to creating automatic patch panel for analog computer", in "Teoriya, matematicheskoye obespecheniye i primeneniye neodnorodnykh vychislitel'nykh sistem: Materialy seminara" [Theory, Software and Application of Heterogeneous Computer Systems: Seminar Materials]. Moscow, December 1973. Moscow, Izdatel'stvo MDNTP, 1973, pages 153-158.

6900
CSO: 1863/64

UDC 681.326.74

METHODS FOR ORGANIZING DIAGNOSIS OF SPECIAL-PURPOSE PROCESSORS AND DEVICES

Kiev GIBRIDNYYE VYCHISLITEL'NYYE MASHINY I KOMPLEKSY: LOKAL'NYYE ASU I USTROYSTVA VYCHISLITEL'NOY TEKHNIKI in Russian No 4, 1981 (signed to press 6 May 81, manuscript received 11 Jan 80) pp 56-59

[Article by R.S. Khalatyan from book "Hybrid Computers and Systems: Local Automatic Control Systems and Computer Devices", edited by G.Ye. Pukhov (editor-in-chief) et al., Izdatel'stvo "Naukova dumka", 1000 copies, 121 pages]

[Text] The extensive use of microprogramming in computing practice has led to the development of diagnostic methods with improved localization capacity. Based on analysis of computer microdiagnostic procedures (microprogrammed diagnostics), the item [1] isolates methods of implementation which use programs of instructions which comprise the instruction system of the computer being diagnosed, as well as special microprograms.

It should be noted that certain assumptions limit application of the methods considered in [1]. For example, the remark in [2] with respect to the implicit assumption that micro-operations can only "disappear" is valid. We might add that the method proposed in [1] for obtaining the set of micro-instructions which makeup a special diagnostic microprogram facilitates obtaining micro-instructions with compatible micro-operations; however, the question of selecting those micro-instructions which would satisfy the "observability" condition defined in [3] still remains unresolved.

The results of [4,5], which examine an "unspooling" procedure which can be used to define a special microprogram, agree with the transportability and observability conditions in [3]. However, these articles examine discovery for an autonomous device. The case of asynchronous devices, for all of which transportability condition must be met in diagnosis, is omitted.

Articles [6-8] are practical examples of microdiagnostics executed by special microprograms for devices interfaced with computers. However, these articles resolve the problems of diagnosing only individual sections, rather than the entire device. For example, [7] develops a flag for loss of data occurring due to loss of synchronization signals for a disc controller.

The development of computer peripherals and increased requirements for repairability make it necessary to work out methods of organizing diagnostics with application to broad classes of interfaced devices. One such class consists of special-purpose processors (SP) for processing data files. Analysis of SP development carried out in [9-16] allows us to isolate the following common structural properties:

-- the presence of communications devices which facilitate two-way communications between the SP and central processor, as well as the arrangement of controlling the exchange of control words and operands with the CP;

-- the presence of two asynchronous microprogrammed controllers, one of which operates with the controller and the other with the arithmetic device;

-- indexed addressing of data read from or written into random-access memory (RAM);

-- data verification;

-- the availability of words for controlling conversion of data read from or written into RAM.

We note that the presence of a buffer makes it possible to combine data process and input, while the bucket-brigade principle allows m data to be processed simultaneously (where m is the number of stages in the brigade), thus providing rapid input.

SP data input begins after servicing the interrupt request sent by the controller to the central processor through the communications device. In addition to the interrupt request, the central processor recognizes the operating status of the SP and its availability.

These features make it possible for the SP diagnostic system to use its MUU [microprogrammed controller] with special microprograms designed for micro-diagnostics. When this is done, the microprograms will be used as diagnostic instructions, allowing the central processor access to these microprograms and writing in RAM the SP responses to the input sets from RAM, producing an interrupt request.

The constant addresses intended for predefined control or check information are often used to write the SP responses in RAM. These may be used continuously for the nondiagnostic operating mode of the SP, e.g., writing the SP responses according to the ASK [expansion not given] address. Then the constant addresses are used in order to avoid using unchecked address formation schemes.

In organizing diagnostics for arithmetic unit schemes using the bucket brigade data processing principle, data is taken from all stages of the brigade, making

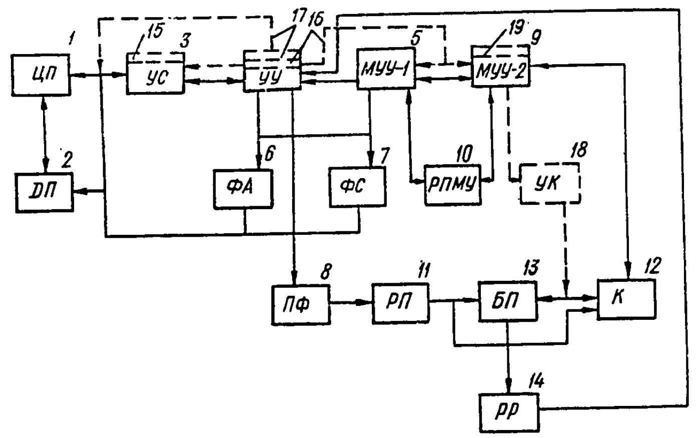


Figure 1.

[Key given in subsequent paragraph -- Tr.]

large combination circuits making up the brigade checkable.

The SP systems testing mode can be augmented with microprograms which are used to test the SP autonomously when the central processor is busy.

When servicing an interrupt request, the central processor sends the SP a diagnostic instruction which enables a microtest microprogram which checks the unchecked part of the SP. Microprogram level, either pre-prepared data needed for the test is read from RAM, or internal test micro-operations are executed, after which there is an interrupt request which flags completion of the microtest. After the interrupt request is serviced, the central processor analyzes the response in order to localize trouble in the section checked by the microtest.

Figure 1 shows the component parts of an SP and the diagnostic hardware, where the dotted line indicates the following devices which operate in the diagnostic mode: 1 -- central processor; 2 -- RAM; 3 -- communications device; 4 -- controller; 5 -- microprogrammed operand and control word controller; 6 -- RAM address formation circuit; 7 -- data verification formation circuit; 8 -- central processor--SP data exchange format converter; 9 -- microprogrammed operand processing controller; 10 -- flag register to coordinate work between the two microprogrammed controllers; 11 -- register for transmitting operands to processor; 12 -- bucket-brigade processor; 13 -- operand and intermediate result buffer;

14 -- result register; 15 -- section of communications device which receives diagnostic instructions; 16 -- section of controller which allows SP to operate in diagnostic mode in case of malfunction; 17 -- section of controller which allows unformed addresses to be used for writing into RAM; 18 -- switching control for transmission of data from bucket brigade stages; 19 -- section of microprogrammed operand processor used to store microprograms which control autonomous diagnostics for processor.

Figure 2 shows the sequence in which the SP diagnostics are executed: 1 -- preparation of operands for diagnostic instructions and area of RAM where SP responses will be written; 1' -- "available" SP state; 2 -- execution of microtest microprogram; 2' -- CP awaits completion of microtest; 3 -- analysis of SP response in order to localize trouble; 3' -- SP "interrupt" status.

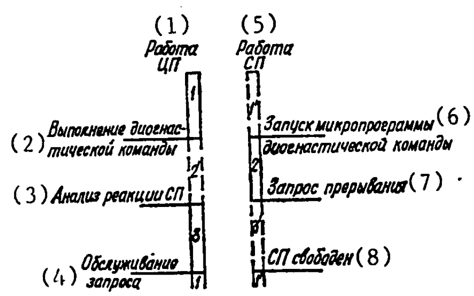


Figure 2.

Key:

1. central processor
2. execution of diagnostic instruction
3. analysis of SP response
4. request servicing
5. operation of SP
6. enable diagnostic instruction microprogram
7. interrupt request
8. SP free

This is a two-level organization of the SP diagnostics. The actual diagnostic procedure and analysis of results are activated at the program level, while the procedure itself occurs at the microprogram level.

It is also possible to obtain a two-level organization for the case in which only the diagnostic procedure is activated at the program level, and the procedure and its analysis are activated at the microprogram level. Then the microprogrammed controller must be able to fetch both micro-instructions and constants (as is done in the "Nairi" machines) and to compare particular bits of the constant used for the standard with the response being tested. With this microprogrammed controller design, trouble can be localized using the indication of the microinstruction address, after which the comparison is made and the operation of the microprogrammed controller stops if the standard and the response do not match. This sort of SP microdiagnostic organization is possible when the microprogrammed controller has enough memory to store the standards. In general, when special microprograms are used for SP microdiagnostics both with and without combination, the problem arises of limiting the amount of microprogrammed controller memory allocated for them.

When special microprograms are used as diagnostic instructions, the diagnostics have the advantage that the SP is tested in real time without changing the machine cycle in either the central processor or the SP.

Real time testing makes it possible to determine such time parameter deviations in the SP operation as the SP--central processor connect time after an interrupt request initiated by the SP is serviced. One example of detecting a violation of synchronization in the operation of two devices in the SP (in other words, the occurrence of a critical state) is the construction of a microprogram which tests the conditions which occur after these devices have operated. In general, the problem of constructing a diagnostic SP microprogram reduces to the problem of accommodating, in the limited microprogrammed controller memory, the diagnostic microprograms of individual SP sections which are diagnosed in a defined sequence (for example, using the "unspooling" method). The diagnostic microprogram designed to control the diagnosis of a particular class of malfunctions in an individual section of the SP together with the hardware associated with this section must have a self-diagnosing capability.

Representing diagnostic microprograms with diagnostic instructions makes it possible to use them in sections which operate under the control of the diagnostic monitor, which makes it easier to control the section. This organization was used in developing the diagnostics for the SP which works in the YeS-1045 computer system. The format of the diagnostic instructions corresponds to the format of the channel instructions used in the Unified Computer System.

In conclusion, we note the possibility of organizing a "closed" procedure (without operator intervention) for diagnosing a peripheral processor or peripheral devices. In doing this, the computer must have an error bit which corresponds to the malfunctioning peripheral which, when the operating system sets a fault flag, fetches the program for diagnosing the malfunctioning device from external storage.

BIBLIOGRAPHY

1. Ramamoorthy, C.V., Chang, L.C. Modeling system and test procedures for microdiagnostic. IEEE Trans. Comput. C, 1972, 21, N 11, p. 1169-1183.

2. Bedeshenkov, V.A., Volkov, A.F. "Automation of designing digital computers with self-diagnosis of malfunctions." In "Diskretnye sistemy: Mezhdunar. simpoz." [Discrete Systems: International Symposium]. Riga, Izdatel'stvo "Zinatne", Vol 2, p 45-54.

3. Dulyaev, V.A., Sivachenko, P.M. "Principles of using microprogrammed testing and diagnosing in small digital computers." UPRAVLYAYUSHCHIYE SISTEMY I MASHINY, No 11, 1975, p 45-50.

4. Kagan, B.M., Mkrtumyan, I.B. "One approach to designing autodiagnostic systems for digital computers." VOPR. RADIOELEKTRONIKI. SER. EVT, No 8, 1971, p 42-51.

5. Inagaki, M. Test and diagnosis program generation using microinstruction. NEC Res. and Develop., 1972, N 26, p 35-52.

6. Pat. 3798613 (USA). Controlling peripheral subsystem/G.H. Edstrom, E.P. Lutter. Publ. 19.03.74.

7. Pat. 3882459 (USA). Deadracking system/G.J. Burlon, D.R. Taylor. Pub. 06.05.75.

8. Pat. 3911402 (USA). Diagnostic block in a computing system/P. Lea. Pub. 07.10.75.

9. Lebedev, A.V., Makarov, V.A., Sorokin, G.K. "Special-purpose processor for processing data files." In "Mnogoprotsessornye vychislitel'nye sistemy" [Multiprocessor Computing Systems]. Moscow, Izdatel'stvo "Nauka", 1975, p 100-109.

10. Gasparayan, L.Kh., Nalbandyan, Zh.S., Tadevosyan, V.A., et. al. "Structural features and functional characteristics of matrix processor in YeS 1045 computer." VOPR. RADIOELEKTRONIKI. SER. EVT, No 10, 1978, p 105-109.

11. Nowlin, R.W., Gustafson, D. A microprogrammed machine architecture for efficient matrix multiplication. In: Ninth Annul. Workshop on Microprogramming. Sept., 1976. New York, 1976, vol 7, no 3, p 56-61.

12. Lynch, W.C. How to stuff an array processor. In: Proc. Third Texas Conf. Comput. Systems, 1974. New York, 1974.

13. Pat. 340611 (USA). Parallel operations in a vector arithmetic computing system/A.D. Falkoff. Publ. 22.05.69.

14. Pat. 3537074 (USA). Parallel operating array computer/R.A. Stokes, G.H. Barnes. Publ. 27.10.70.

15. Pat. 3541516 (USA). Vector arithmetic multiprocessor computing system/ D.N. Senzing. Publ. 27.10.70.

16. Pat. 3775753 (USA). Vector order computing system/W.D. Kastner. Publ. 27.11.73.

6900
CSO: 1863/64

UDC 681.33

SYNTHESIS OF OPTIMAL DIGITAL-ANALOG REGULATOR FOR CONTROLLING THERMAL OBJECT

Kiev GIBRIDNYYE VYCHISLITEL'NYYE MASHINY I KOMPLEKSY: LOKAL'NYYE ASU I USTROYSTVA VYCHISLITEL'NOY TEKHNIKI in Russian No 4, 1981 (signed to press 6 May 81, manuscript received 10 Jan 80) pp 99-102

[Article by G.V. Yevstratov from book "Hybrid Computers and Systems: Local Automatic Control Systems and Computer Devices", edited by G.Ye. Pukhov (editor-in-chief) et al., Izdatel'stvo "Naukova dumka", 1000 copies, 121 pages]

[Text] The investigation of heat- and mass-transfer systems requires solving analysis and synthesis problems. In solving analysis problems, the stability and performance of the system in question must be determined. Since the main tasks of design are to select optimal object parameters (parametric optimization) and to synthesize a system for controlling the object, the synthesis problem consists essentially of selecting the structural diagram of the system, its parameters and technical implementation such that the required control performance is provided.

The methods used to control objects depend upon the available data. Since it is impossible to obtain complete information on the status of heat- and mass-transfer processes under actual conditions, the problem of synthesizing optimal control of these processes with incomplete measurement is also urgent when all types of limitations are manifested in the optimal process, in terms of both the output coordinates of the object and the control action. The latter includes problems involved in investigation and synthesis of thermostatting systems.

The Department of Automation and Telemechanics at Khar'kov Polytechnical Institute imeni V.I. Lenin has been working for several years on synthesizing optimal control in heat- and mass-transfer systems with incomplete measurement, and on solving direct and inverse thermal conductivity problems which are very close to optimization problems in three directions.

The first direction in solving these problems using RC-network models which implement a numerical method is universal. It can be used to model boundary problems of field theory in any statement for bodies with complex boundaries with sufficient accuracy, and is in terms of many indicators competitive with digital computers. Since the electrical time in RC-network integrators is many times smaller than the actual thermal time, comprising a fraction of a second, RC-models are automated. This favorable feature reduces the labor involved in reaching a solution. One shortcoming of this method is the increased complexity of the hybrid computer system and its reduced reliability as the requirements for solution accuracy increase.

The second direction of solving such problems encompasses a narrower group of objects for which, allowing for their singularities and using several insignificant simplifications and the choice of a coordinate system, it is possible to find a mathematical model which describes that object with the required accuracy. Furthermore the mathematical model is sought in the form of a system of algebraic equations which is suitable for solution by computer using numerical methods, or which can be solved analytically. This approach was used in [1] for the optimal arrangement of heaters along the surface of a chamber, and for using them as the basis for synthesizing a hybrid computer system for solving optimization problems and direct and inverse field theory problems; of the thermostat with assigned limitations on the temperature drops across the thermostat chamber [sic -- Tr.].

The third direction in solving optimal control problems is examined in [2,3].

A number of qualitative and quantitative criteria are used to select a solution method and means for its numerical implementation. As was shown in [4], the conclusion of the advantage of numerical methods should not impede the development of analytical methods, which have the undisputable advantage that they can be used to obtain a numerical solution at a given point at a given instant. If an analytical solution can be implemented by computer, and if the accuracy, cost and time of the solution are comparable to the analogous values for a solution obtained numerically, the analytical solution is preferable.

Let us consider the synthesis of a digital-analog system for solving optimal control problems in more detail.

Figure 1 shows the functional diagram of a hybrid computer system for investigating RP-objects [expansion not given]. This setup includes the following basic elements: digital-analog network model -- RC-network; boundary and initial condition assignment section; time interval formation section; digital control and readout device; data measurement and display device and regulator model which, in turn, consists of the regulator itself, a reference-input element, actuator, repeater and sensor.

The structure and operation of sections (5), (7), (8), (9), (10) and (11) is described sufficiently completely in [5,6,8].

The main element in the hybrid computer system is the passive RC-network, which serves as the dynamic model-analog of the RP-object of control. This network

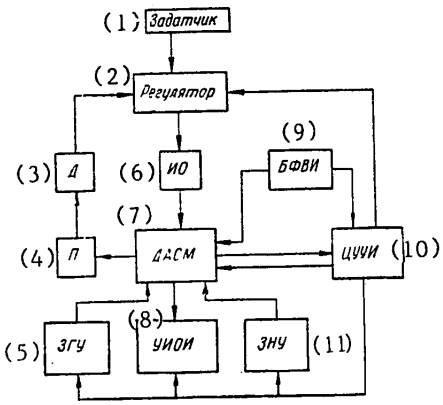


Figure 1.

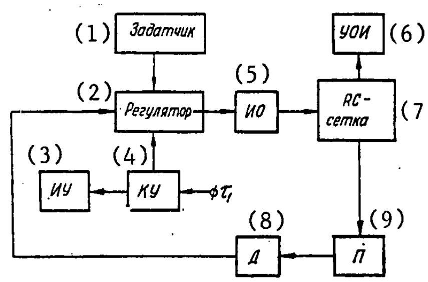


Figure 2.

Key:

1. reference-input element
2. regulator
3. sensor
4. repeater
5. boundary condition assignment section
6. actuator
7. discrete-analog network model
8. data measurement and display device
9. time interval formation unit
10. digital control and readout device
11. initial condition assignment section

1. reference-input element
2. regulator
3. IU [expansion not given]
4. correction device
5. actuator
6. data display device
7. RC-network
8. sensor
9. repeater

integrates the equations which describe the dynamics of the thermal field in the area in question. As a result of integrating the original equation, the RC-network forms the current values of certain components of the state vector of the controlled object. For example, the RC-network defines the nonstationary thermal state of elements of the construction when entering the thermostat mode.

The boundary condition assignment section or actuator of the regulator produce the inputs -- currents and voltages -- applied to the RC-network; these serve as the analog of the controlling thermal action.

Let us consider a special case which is fairly important and often encountered in applications: this is the case of optimal control, in terms of speed, of an RP-object when the end state of the object is reached at the maximum value of the control action, and arbitrary limitations are placed on its output coordinates and on the control actions. This condition is characteristic for transient operating modes of most thermal objects. This causes stabilization sectors to occur in the control which correspond to movement of the controlled object along the boundary of the region of acceptable parameters. The control has maximum value everywhere (except for special sectors), i.e., it is on the boundary of the region of acceptable states. This feature of this class of optimal control problems makes it possible to obtain solutions using a hybrid computer system with the proposed structure.

In order to obtain a solution to the optimal problem, the operation of the system is organized as follows. If the state vector is located within the Ω_y-domain of acceptable parameters, the regulator allows the maximum tolerable control actions to be output to the RC-network. As soon as the boundary of the Ω_y-domain is reached, the regulator corrects the GU [expansion not given] such that the maximum values of the monitored parameters are not exceeded. Thus, the problem of the synthesized optimal regulator is to organize movement of the model-analog of the controlled object along the boundary of the region of acceptable values. In solving this problem, the regulator and the DASP [expansion not given] represent a closed system which keeps the output coordinates of the controlled object at acceptable levels.

Figure 2 shows the functional diagram of an optimal regulator with manual optimal solution search. Manual optimal solution search in a closed control system assumes the presence of an operator. The person using the data display device to observe the transient process on the object keeps the output coordinates of the object at acceptable levels. The operator does this by means of the correction device, which acts upon the regulator for τ_{zap} starting with output of pulse τ_1 (decision time) from the digital control and display device (cf. Figure 1). For time τ_{zap} the regulator outputs maximum control action to the actuator and then to the object. After that, ordinary (e.g., proportional) regulation begins, which optimizes the error.

The operator's task is now to use the model with the help of the correction device to find a duration τ_{zap} experimentally such that the curve of the transient process is optimal.

The actuator in a thermostatting system is a controllable reversible thermopile in which, depending upon the magnitude and sign of the input current, the thermopile isolates heat or cold with the appropriate magnitude [sic -- Tr.].

The electrical analog of the thermopile is a controllable reversible stable-current source operating into a grounded load (Figure 3). The stable current source uses a single 1UT402A integrated circuit, which is an operational amplifier with simultaneous negative and positive feedback. Its output current depends upon the input voltage

$$I_n = \frac{U_{in}}{R_2}.$$

The regulator implements the following control principle::

$$P = P_{max} = const, \ 0 \leq t \leq \tau_{zap},$$

$$P = \varepsilon k_p k_t = var, \ t > \tau_{zap},$$

where t is time; τ_{zap} is delay time in connecting negative feedback, i.e., the amount of time the thermopile operates at maximum power P_{max}; ε is the regulation error; k_p is the regulator gain; k_t is the gain of the thermopile. Time τ_{zap} is measured by the IU (cf. Figure 2).

Figure 4 shows the functional diagram of an optimal regulator which automatically finds an optimal solution. Here information about the state of the object is continuously input to the limitation section (8). If even one of the state parameters of the object of control is too high, the limitation section acts upon the correction device with memory (7) which reduces τ_{zap} by one step.

During the next solution period τ_1 the process is repeated, continuing until the state parameters of the object exceed their bounds.

Since an RC-network hybrid computer system is a repetitive-solution model, it can easily be used to implement this iterative method of solving these problems. The use of iterative methods eliminates dynamic error and increases the accuracy of the sought results.

Research has indicated that the solution to a problem on a search-type device is obtained as the result of n iterations. However, the total solution time due to search does not increase significantly. This is because of the high speed of an RC-network hybrid computer system.

In conclusion, we note that the structure of this hybrid computer system allows it to be used to solve more complicated control problems, including those which are characterized by the presence of several output coordinates of the object which have different simultaneous limitations.

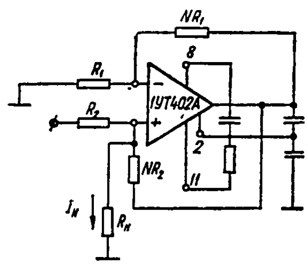


Figure 3.

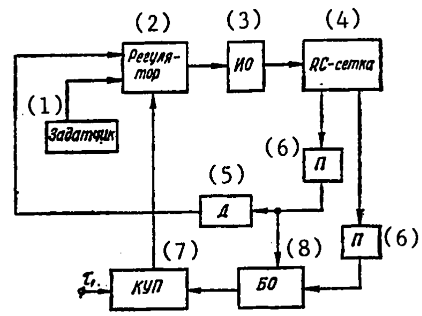


Figure 4.

Key:

1. reference-input element
2. regulator
3. actuator
4. RC-network
5. sensor
6. repeater
7. correction device with memory
8. limitation section

BIBLIOGRAPHY

1. Yevstratov, G.V., Prokof'yev, V.Ye. "Mathematical model of temperature field and optimization of placement of heaters along surface of thermostat chamber." In "Teplomassoobmen i modelirovaniye v energeticheskikh ustranovkakh" [Heat- and Mass-Exchange and Modeling in Power Installations]. Tula, Tula Polytechnical Institute, 1979, Part 1, p 142-143.

2. Yevstratov, G.V., Prokof'yev, V.Ye. "Use of mathematical modeling methods to identify and optimize objects with distributed parameters." In "Teoriya i metody matematicheskogo modelirovaniya" [Theory and Methods of Mathematical Modeling". Moscow, Izdatel'stvo "Nauka", 1978, p 110-111.

3. Biryukova, G.V., Yevstratov, G.V. "Approximation of Dynamic Characteristic of Object with Distributed Parameters on RC-Network Model." Present collection, page 3-6.

4. Kozdoba, L.A. "Resheniya nelineynykh zadach teploprovodnosti" [Solutions to nonlinear thermal conductivity problems]. Kiev, Izdatel'stvo "Naukova dumka", 1976.

5. Azarov, G.N., Andriyevskiy, V.M., Garmash, V.V., et.al. "Special-purpose device for automating investigation of thermostatic systems." In "Lokal'nye avtomatizirovannye systemy avtomatiki" [Local automated automation systems]. Kiev, Izdatel'stvo "Naukova dumka", 1978, p 10-18.

6. Positive decision on application No. 2504609/18-24 (085553) on patent (USSR) of 30 December 1977. "Device for modeling closed distributed control systems." G.N. Azarov, V.M. Andriyevskiy, V.V. Garmash et.al.

7. Aleksandrovskiy, N.M., Yegorov, S.B., Kuzin, R.Ye. "Adaptivnye sistemy avtomaticheskogo upravleniya slozhnymi tekhnologicheskimi protsessami" [Adaptive Automatic Control Systems for Complex Technological Processes]. Moscow, Izdatel'stvo "Energiya", 1973.

8. Yevstratov, G.V. "Time-interval formation section for digital-analog network processor." In "matematicheskoye modelirovaniye i gibridnaya vychislitel'naya tekhnika" [Mathematical Modeling and Hybrid Computer Techniques]. Kuybyshev, Kuybyshev Polytechnical Institute, 1977.

6900
CSO: 1863/64

OPTICAL PROCESSING

HOLOGRAPHY AND OPTICAL PROCESSING OF INFORMATION: METHODS AND APPARATUS

Leningrad GOLOGRAFIYA I OPTICHESKAYA OBRABOTKA INFORMATSII: METODY I APPARATURA in Russian 1980 (signed to press 31 Dec 80) pp 2, 235-236

[Annotation and table of contents from the collection "Holography and Optical Processing of Information: Methods and Apparatus", edited by V. G. Skrotskiy, B. G. Turukhano and N. Turukhano, Izdatel'stvo Leningradskogo instituta yadernoy fiziki, 500 copies, 237 pages]

[Text] The 12th All-Union School on Holography was held according to the plan of the Council on Holography attached to the Presidium of the USSR Academy of Sciences and was organized by the Moscow Order of the Red Banner of Labor Physicotechnical Institute.

The following institutions participated in preparation and publication of the proceedings of the 12th All-Union School on Holography: the Leningrad Institute of Nuclear Physics imeni B. P. Konstantinov, USSR Academy of Sciences, the Moscow Order of the Red Banner of Labor Physicotechnical Institute and the Order of Lenin Physicotechnical Institute imeni A. F. Ioffe, USSR Academy of Sciences.

The 12th All-Union School on Holography was held at Pasanauri.

Leading Soviet scientists and engineers assembled at the All-Union School to discuss the latest advances in holography, quantum electronics and recording media for recording holograms.

The proceedings of the 12th All-Union School on Holography are of interest for a wide range of scientists and engineers specializing in the field of holography.

The organizing committee of the school is grateful to the participants who read such interesting reports and presented them for publication in this collection.

Contents

6521
CSO: 1863/26

UDC 778.38

RECORDING AND PROCESSING OF MODULATED OPTICAL SIGNALS

Leningrad GOLOGRAFIYA I OPTICHESKAYA OBRABOTKA INFORMATSII: METODY I APPARATURA in Russian 1980 (signed to press 31 Dec 80) pp 86-92

[Article by V. A. Zubov and A. V. Krayskiy from the collection "Holography and Optical Processing of Information: Methods and Apparatus", edited by V. G. Skrodskiy, B. G. Turukhano and N. Turukhano, Izdatel'stvo Leningradskogo instituta yadernoy fiziki, 500 copies, 237 pages]

[Text] The use of a transient reference wave with linear variation of frequency through the cross-section permits one to achieve total recording and processing of a modulated optical signal. These problems were considered theoretically for monochromatic radiation in [1, 2]. Experiments on recording this signal are considered in this paper and the operation of schemes that permit one to find the structure of the signal and its spectrum with resolution considerably exceeding the width of the laser spectrum and which reaches fractions of a hertz are considered in this paper.

A diagram of a two-channel experimental installation is presented in Figure 1. Laser emission after diaphragm D1, which limits the homogeneous section of the wave, is divided into two parts by means of a prism PR. A transient wave having the following form in the recording plane is shaped in one channel formed by diaphragms D2 and D3 and the telescopic system L1 and L2 with quasi-linear motion of the components of L1 [3]:

$$E_1(x,t)=\sum_{k=1}^{N} \mathcal{E}_{1k} \exp\left[-i\omega_k t + i a_k(t) + i\frac{\omega_k v}{c f_2} x t\right] \quad (1)$$

where K is the number of the specific radiation mode, the total number of modes i N, ε_{ak} is amplitude, ω_k is the mean frequency and $a_k(t)$ is the random phase of this mode. The second channel formed by the telescopic system L3, L4 and mirror 3 is the working channel. The signal is modulated in the mean focal plane of this telescopic system. The effect of the object is modulated by an interrupter disk M in the experimental circuit. Part of the emission of the working channel is deflected by a semitransparent plate to photodetector FEU and is further recorded on an oscillograph OSTs so as to have information about the amplitude modulation of the signal being recorded. The radiation from the working channel impinges on the recording plane xy and the direction of propagation is described by a cosine $\cos\theta_{xs}$. The field from the object has the form:

$$\mathcal{E}_o(x,t)=\sum_{s=1}^{N}\mathcal{E}_{os}\,\varepsilon_o(t)\exp\left[-i\omega_s t+i a_s(t)+i\frac{\omega_s}{c}x\cos\theta_{xs}\right], \quad (2)$$

where s is the number of the individual laser emission mode and $\varepsilon_0(t)$ is the complex amplitude that describes signal modulation. If ordinary conditions are fulfilled, a hologram with amplitude transparency consisting of two components is produced: a component related to a constant glow from two channels and a component with information about the signal. This hologram can be exposed within a limited time T since optical system L1 and L2 should operate in paraxial approximation when the reference wave is formed. As a result of averaging the information, only terms with s = k make a nonzero contribution during recording. As a result we have

$$t(x)\sim\int_{-T/2}^{T/2}\varepsilon_o(t)\sum_{k=1}^{N}\mathcal{E}_{ok}\,\mathcal{E}_{rk}^{*}\exp\left[i\frac{\omega_k}{c}x\cos\theta_{xk}-i\frac{\omega_k V}{c f_2}xt\right]dt\ + \quad (3)$$

+ complex integration

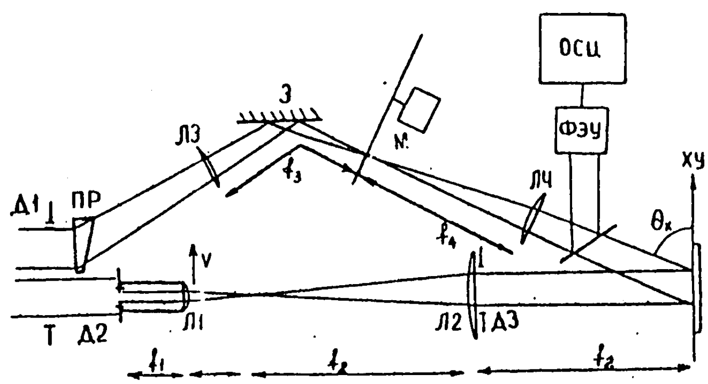


Figure 1. Diagram of Experimental Installation for Recording Modulated Optical Signals

Upon restoration the hologram G is illuminated by a plane laser emission wave (Figure 2) consisting of M modes and the mean direction of propagation is given by a cosine--$\cos\theta_{xm}$

$$\mathcal{E}_p(x,t)=\sum_{m=1}^{M}\mathcal{E}_{pm}\exp\left[-i\omega_m t+i b_m(t)-i\frac{\omega_m}{c}x\cos\theta_{xm}\right], \quad (4)$$

where m is the number of the mode. All calculations were made in Fresnel approximation.

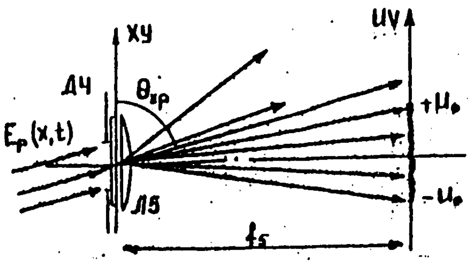


Figure 2. Diagram of Restoring the Structure of a Modulated Signal

Optical system L5 performs the operation of Fourier transformation in the restoration phase. Let us take into account that $|\omega_m - \omega_k| \ll \omega_m, \omega_k$ and the direction cosines can be selected so that $|\cos\theta_{xm}| = |\cos\theta_{xk}|$; then for a hologram with dimensions equal to $2x_0$ along the x axis, the spread function has a simple form:

$$\operatorname{sinc}\left[x\frac{\omega_k}{c}\left(\frac{vt'}{f_2}+\frac{u}{f_5}\right)\right]. \tag{5}$$

With regard to the final resolution of the field component that produces the pattern of the recorded signal in the rear focal plane UV of the optical system, there will be

$$E(u,t)\sim\sum_{m=1}^{M}\exp\left[-i\omega_m t+i\phi_m(t)\right]\tilde{E}_0\left(-\frac{f_2 u}{f_5 v}\right), \tag{6}$$

i.e., the recorded modulated field is restored in three-dimensional representation. Let us determine resolution by the distance between the zeros of apparatus function (5), which yields

$$\delta t\sim\frac{2\pi c f_2}{\omega_k x_0 v}\sim\frac{\lambda f_2}{x_0 v}, \qquad \delta u\sim\frac{2\pi c f_5}{\omega_k x_0}\sim\frac{\lambda f_5}{x_0}. \tag{7}$$

The position of the restored image is given by the maximum spread function for moments of time $t = \pm T/2$ and its dimensions and number of resolved components are equal to

$$2u_0 = f_5 v T/f_2, \quad n = T/\delta t = 2u_0/\delta u = \frac{vTx_0}{\lambda f_2}. \tag{8}$$

The experiment was performed with the following values: $\lambda = 0.63\cdot10^{-3}$ mm, T = 6.8 seconds, $v \simeq 2$ mm/s, f_2 = 300 mm, f_5 = 300 mm and $2x_0 \simeq 8$ mm. Calculation yields: $|\cos\theta_{kr}| \le 0.023$, $2u_0 \simeq 14$ mm, $\delta t \simeq 25$ ms, $u \simeq 0.05$ mm and $n \simeq 280$. The experimental results for different modulation frequencies and different on-off time ratios are presented in Figure 3.

The scheme for finding themodulated signal spectrum $F(\Omega)$ using steady restoring wave (4) is presented in Figure 4. The field in observation plane xy, optically conjugate with the hologram G, based on (3) within the conditions used in derivation of (5), has the form

$$\begin{aligned} E(X,t) &\sim \sum_{m=1}^{M} \exp[-i\omega_m t + i b_m(t)] \int_{-T/2}^{T/2} E_0(t') \exp\left[i \frac{\omega_K}{c} \frac{\rho_1 v}{\rho_2 f_2} X t'\right] dt' \sim \\ &\sim \sum_{m=1}^{M} \exp[-i\omega_m t + i b_m(t)] \tilde{F}\left(\frac{\omega_K \rho_1 v}{c \rho_2 f_2}\right), \end{aligned} \quad (9)$$

where the tilda sign corresponds to convolution of the signal spectrum with the spread function. For three-dimensional resolution in the recording plane, based on (9) and (8), we find

$$\delta X \sim \frac{2\pi c}{\omega_m} \frac{\rho_1}{u_0} \sim \frac{2\lambda \rho_1 f_2}{f_5 v T}. \quad (10)$$

To estimate the spectral resolution, let us make use of the concept $\varepsilon_0(t')$ in the form of a Fourier transform $F(\Omega)$, which yields the spread function

$$\operatorname{sin c}\left[\frac{T}{2}\left(\Omega + \frac{\omega_K \rho_1 v}{c \rho_2 f_2} X\right)\right]. \quad (11)$$

An estimate of the width of the spread function is made similar to (7) and yields $\delta\Omega = 4\pi/T$. It should be noted that from the viewpoint of spectral resolution it is more correct to proceed from the Rayleigh criterion, i.e., $\delta\Omega_p = \delta\Omega/2 = 2\pi/T$ or, converting from circular to orginary frequencies

$$\delta\nu_p = 1/T \quad \text{and} \quad \delta X_p = \lambda \rho_2 f_2 / \rho_1 v T. \quad (12)$$

Thus, spectral resolution is determined by the total recording time. The dispersion range is determined by the frequency range recorded on the hologram, which is equal to

$$\Delta\omega \sim \frac{\omega_K v}{c f_2} 2x_0 \quad \text{or} \quad \Delta\nu \sim \frac{\Delta\omega}{2\pi} \sim 2\frac{v x_0}{\lambda f_2}. \quad (13)$$

Dispersion in the recording plane (see (12)) and the number of resolved components (see (13) and (12)) are equal to

$$\delta\nu_p/\delta X_p \sim v\rho_1/\lambda\rho_2 f_2, \quad n = \Delta\nu/\delta\nu_p = 2\frac{v}{\lambda f_2} x_0 T. \quad (14)$$

The linear dimensions of the spectrum are determined by the maximum spread function for three-dimensional resolution at $x = \pm x_0$, which yields

$$2X_0 = 2x_0\rho_2/\rho_1. \quad (15)$$

With regard to the values used in the experiment (ρ_1 = 125 mm, ρ_2 = 260 mm and f_5 = 85 mm), we find the following estimates: $\delta\Omega_p \simeq 0.9\ s^{-1}$, $\delta\nu_p \simeq 0.15$ Hz, $\delta X_p \simeq 0.03$ mm, $\Delta\nu \simeq 85$ Hz, $\delta\nu_p/\delta X_p \simeq 5$ Hz/mm, $2X_0 \simeq 16.5$ mm and $n \simeq 550$.

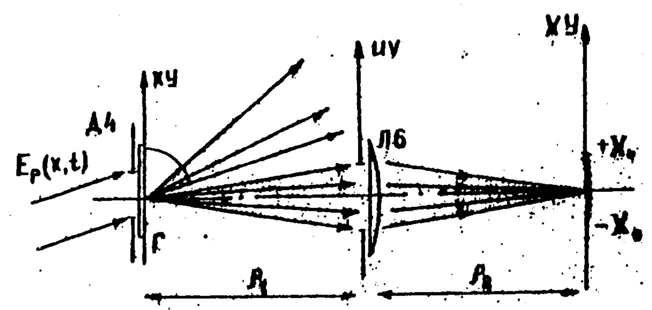


Figure 4. Diagram for Finding the Spectrum of a Modulated Optical Signal

The experimental restuls are presented in Figure 3. As was noted, the effect of an object on the signal was modulated by modulation at frequency of 1.03 and 2.06 Hz with different on-off time ratios. Calculation of the modulation frequency from the spectrograms yields 1.0 and 2.0 Hz. Illustrations show that the real resolution considerably exceeds 1 Hz.

BIBLIOGRAPHY

1. Zubov, V. A., A. V. Krayskiy and T. I. Kuznetsova, PIS'MA ZHETF, Vol 13, 1971.

2. Zubov, V. A., KRATKIYE SOOBSHCHENIYA PO FIZIKE, No 7, 1972.

3. Borkova, V. N. and V. A. Zubov, Proceedings of a Seminar "New Developments in the Field of Optical Holography and Their Industrial Use," 21 pages, Leningrad, Izd-vo LDNTP, 1979.

6521
CSO: 1863/26

UDC 550.834

HOLOGRAPHIC VISUALIZATION OF UNDERGROUND OBJECTS

Leningrad GOLOGRAFIYA I OPTICHESKAYA OBRABOTKA INFORMATSII: METODY I APPARATURA in Russian 1980 (signed to press 31 Dec 80) pp 150-155

[Article by A. V. Zuyevich, V. V. Alekseyenko and V. B. Gavryushin from the collection "Holography and Optical Processing of Information: Methods and Apparatus", edited by V. G. Skrotskiy, B. G. Turukhano and N. Turukhano, Izdatel'stvo Leningradskogo instituta yadernoy fiziki, 500 copies, 237 pages]

[Text] The results of experiments on holographic visualization of underground test objects are reported. The experimental installation is described. Holograms and restored images of test objects are presented. The design features of a marine holographic system are discussed.

The past decade has been characterized by active development of the continental shelf of the seas, with which the main hopes for solving the problem of expanding the country's raw materials base are now linked. Development of the shelf assumes solution of a number of problems of both an exploratory and commercial nature. One of these problems is detailed study of the upper part of the sedimentary mantle. The need for this is related to problems of constructing stationary drilling platforms, laying oil and gas pipelines and construction of underwater storage tanks. Moreover, solving the problem of search for and identification of objects at the bottom require development of means of visualizing seismoacoustic images.

One of the most promising directions in solving the indicated problems are holographic methods. An experimental check of long-wave holography systems has been made quite recently, but mainly under laboratory conditions [1-3]. It was feasible in this regard to check the capabilities of holographic systems under conditions approaching real conditions. This check was made in a test area on an artificial reservoir. The test objects were located on the bottom or under the bottom of the reservoir. The emitter was attached to a raft and was made immobile during the experiment. The emitter itself was assembled on the basis of a piezoelectric cylinder (working frequency f - 10 kHz) and was equipped with a shaper to create a main lobe radiation pattern of approximately 30°. Radio pulse emission with pulse length of 1.0-2.0 ms was used. The framework together with the receiving antenna and part of the processing apparatus was moved along rails laid along the basin. The receiving antenna contained 128 channels and the synthesized (linear) dimension was approximately 42λ. The channels were interrogated during movement of the antennas at a frequency of 250 kHz along the antenna and 50 Hz in the direction of motion. The interrogation signal was fed by cable to the processing system located on shore. The reference signal was simulated in an electronic channel and the radiation pulse parameters and readout mode were controlled from the control console of the processing apparatus. The information to be recorded was stored on the target of a ZELT [Shielded cathode-ray tube]. In this case movement of the line along the frame was synchronized with the movement of the antenna within the aperture to be synthesized. A hologram which was subsequently visualized on the ELT [Cathode-ray tube] screen was formed on the target upon completion of the synthesis cycle. The hologram was then photographed from a distance of 1.5-2 meters for the appropriate reduction, processed and the reduced copy measuring from 0.6 X 0.6 to 1.0 mm^2 was restored in the light of a He-Ne laser. The restored image was observed in a microscopic system.

The symbols "S" and "H", made of sheet steel 1 cm thick, and also a pyramid of marble cubes were used as the test objects. The dimensions of the "S" and "H" symbols were 2.0 X 1.5 m^2 (13 X 10 λ^2) (the width of the letters was 0.45 m = 3 λ), while the side of the marble cube was 0.4 m or approximately 3 λ. The "S" and "H" objects were located under the bottom one over the other so that the symbol "H" was at a depth of 3 meters while symbol "S" was 2 meters above it and 1 meter from the bottom. The vertex of the marble pyramid was located 1 meter from the surface.

The shape of the objects, the recorded holograms and the restored images are presented in Figure 1. The images are quite acceptable with regard to the complexity of the medium and the smallness of the parts of the object, although not of very high quality, which makes it possible to identify the objects with adequate confidence.

The experiments showed that these systems permit one to solve the problem of search and identification of objects in soils under marine conditions. This made it possible to turn to developing a marine version of the holographic complex. The complex is designed to solve problems of investigating the bottom and the bottom sedimentary mass or rather its first few tens of meters under conditions of the continental shelf at water depth of 100-150 meters. In this case the given complex permits one to visualize (i.e., to obtain visual images of) objects of different nature located in the bottom sedimentary mass.

The marine holographic complex now developed includes an aggregate of apparatus assemblies and technical means of registration, recording, processing and visualization of the holographic information on board a ship under autonomous navigation.

The complex consists of a ship on which the information recording, processing and visualization apparatus is located and a towed device. The towed device is designed to study the sounding signals, to record the signals reflected by the objects by the two-dimensional aperture synthesizing method and preliminary processing of the received signals.

Electronic apparatus that records and processes the information is located on the ship, while the final result of processing is an acoustic hologram visualized on an ELT screen. A visualized hologram photorecording unit and device for optical restoration of holograms with a He-Ne laser are located on the same ship. Moreover, a sounding signal power amplifier is located on the ship, while the ship itself is equipped with a velocity measuring device required for proper recording of holograms. The latter condition is related to the need for strict geometric conformity between the position of the antenna within the aperture to be synthesized and the position of the information to be imaged on the ELT screen within the frame. The towed device of the complex consists of a receiving antenna of the preliminary processing unit of the damper-sinking emitter of the equipment system designed for towing and loading-unloading by the towed system.

The receiving antenna is a hydroacoustic device having the shape of a wing in cross-section, which provides the required hydrodynamics during towing. The antenna consists of preassembled sections, each of which contains rudder-stabilizers, while the outer sections are also equipped with rudders-sinking units. A total of 128 hydrophones connected to amplifiers are located on the bottom of the antenna in its middle part. The leads of the amplifiers are connected by cable to the preliminary processing unit.

The preliminary processing unit is made in the form of a cigar-shaped gondola and contains the electronic assemblies (phase detectors and decoder) for processing the signals coming from the receiving components by cable.

The emitter is designed to irradiate the objects of investigation. The emitter itself is made on the basis of cylindrical piezoceramics and is equipped with a radiation pattern shaper. The skin of the emitter is made from an acoustically strong material and mainly performs the function of providing the hydrodynamics required for towing. Both gondolas are firmly connected to each other and are attached to the antenna by two metal rods. The antenna together with the gondolas has low positive buoyancy. The outboard part of the system is towed by a tow cable to which the damper-sinking unit is attached, that ensures towing safety with sharp jerks of the rope. The cable is used to supply power to the outboard electronic units, to control the operation of these units, to gather information and also to excite the emitter. The equipment system includes a swivel, lowering-raising system and a spring hook. The parameters of submerging the antenna and stabilization of its course are varied manually in the given version, while the parameters themselves are selected empirically. Foam plastic floats designed to stabilize the depth of submersion are attached to halyards on the edges of the antenna.

Tests of the running qualities of the outboard part of the complex, conducted in 1979 in a basin of the Sea of Azov, showed its complete suitability for operating under maritime conditions by the long-wave holography method. The complete complex will be tested in the near future. These experiments make it possible to determine the capabilities of the long-wave holography method in solving problems of marine geology and geophysics and consequently to determine the place of holographic systems in solving problems of exploiting the mineral resources of the world ocean.

BIBLIOGRAPHY

1. Meterel', L. F., "The Comparative Importance of Phase and Amplitude in Acoustic Holography," "Akusticheskaya golografiya" [Acoustic Holography], translated from English, Leningrad, Sudostroyeniye, 1975.

2. Farr, J., "Experiments in Acoustic Holography Using Computers," "Akusticheskaya golografiya", Translated from English, Leningrad, Sudostroyeniye, 1975.

3. Zuyevich, L. V., V. V. Alekseyenko and V. M. Sugak, "Recording and Restoration of Long-Wave Holograms in Small Apertures," PIS'MA V ZHTF, Vol 4, No 6, 1978.

6521
CSO: 1863/26

UDC 535

PROCESSING SEISMIC INFORMATION WITH COHERENT OPTICAL SYSTEM

Leningrad GOLOGRAFIYA I OPTICHESKAYA OBRABOTKA INFORMATSII: METODY I APPARATURA in Russian 1980 (signed to press 31 Dec 80) pp 156-163

[Article by K. B. Gendovich and K. S. Stoyanova-Pushkarova, Peoples Republic of Bulgaria, from the collection "Holography and Optical Processing of Information: Methods and Apparatus", edited by V. G. Skrotskiy, B. G. Turukhano and N. Turukhano, Izdatel'stvo Leningradskogo instituta yadernoy fiziki, 500 copies, 237 pages]

[Text] Some experimental applications of coherent optics and holography methods are considered for processing seismic information. A coherent optical-electronic system was used for spectral-correlation processing of seismic time profiles containing a large volume of information (approximately 10^7 bits). An autocorrelogram and autocorrelation functions were found for the root or group of roots of the seismic profile selected by the interpreter. The effect of attenuation of short waves was estimated by the "24-fold addition" method and the unified statistical characteristics of the factors that determine the capability of separating single waves to investigate the seismic profile was found.*

Problems of search and study of mineral deposits, solution of which is impossible without the participation of computers, have continuously become complicated during the past few years. Most computers that are used, distinguished by low internal storage and relatively low speed, are incapable of operationally processing the continuously increasing volume of incoming information.

One of the methods of progress in the field of processing seismic information is to coherent-optical computers characterized by high speed, economy and flexibility in variation of the program-algorithmic complex [1].

The purpose of the experiments is to obtain autocorrelograms of the seismic profile and also the autocorrelation functions of its individual sections.

The investigation was conducted by the total deep point method (MOGT) from a seismic profile of the Black Sea basin. The seismic profile was processed by the

*The report was made at the 11th All-Union School on Holography.

"24-fold addition" algorithm to reduce the interfering effect of reverberation caused by the water layer and the effect of multiple reflections produced from some reflecting boundaries in the geological structure of the region. As a result a time (seismic) profile consisting of 384 tracks (channels) with four seconds of recording of each channel represented graphically by the variable width method was obtained. The length of the seismic profile in the X direction on the profile was 9.6 km. Interfering multiple waves create imaginary reflecting boundaries on the seismic profile, i.e., on the time image of the geological profile, which make difficult both correct running of the following algorithm for digital machine processing and clear interpretation of the results.

The effectiveness of the mentioned algorithm can be determined directly from the autocorrelogram and autocorrelation functions of individual sections but the type and number of the reverberation systems, their attenuation factors, tectonic disturbances and structural complications in the investigated region are determined from the ratio of amplitudes of the secondary maximums and the main maximum of the autocorrelation function.

The optical-electronic system used is shown in Figure 1, where 1 is a laser, 2 is a filter-collimating unit, 3 is microscopy of the seismic profile, 4 is an astigmatic lens, 5 is a semitransparent mirror, 6 is a television camera, 7 is a monitor, 8 is a storage digital oscilloscope, 9 is a microcomputer and 10 is a photorecorder.

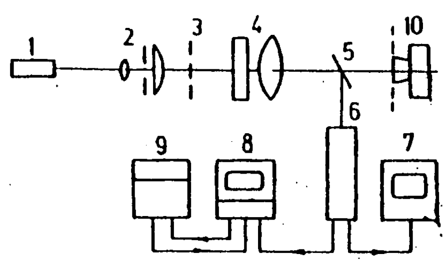


Figure 1. Optical-Electronic System for Producing Autocorrelogram and Autocorrelation Functions: 1--laser; 2--filter-collimating unit; 3--microscopy of seismic profile; 4--astigmatic lens; 5-semitransparent mirror; 6--television camera; 7--monitor; 8--storage oscilloscope; 9--microcomputer; 10--photorecorder

The original seismic profile is reduced by a factor of 18.7. The correct position of the components in the optical system are monitored by means of a special scale built into the pattern of the seismic profile prior to microfilming.

The square of the modulus of the microscopic spectrum is recorded in the spectral plane of the optical system [2]. The positive microfilm produced in the next phase is installed at the input of the system at the microscopic position. The desired autocorrelogram of the seismic profile is found in the spectral plane of the optical-electronic system. The spectrum and autocorrelogram are shown in Figures 2 and 3, respectively. An image of the autocorrelogram can be observed on the monitor by means of the television camera. Some changes were made in the camera to obtain autocorrelation functions so that the light field impinging on the camera tube had the capability of scanning in sequential lines. The digital delay unit of the storage oscilloscope was used to select the camera tube scanning line. The video signals obtained from line scanning are converted to digital form and are recorded in the memory of the microcomputer and in this case the scanning aperture can be formed with sequential addition of adjacent television scanning lines. The autocorrelation functions of the track or group of tracks of the seismic profile selected by the interpreter are observed on the oscilloscope screen (Figure 4).

The experimental method of obtaining the autocorrelogram excludes the necessity of precise adjustment of the filter-hologram in the Van der Luchta method, as does precise setting of the input object and filter in the front focal planes of the first and second lenses, respectively. At the same time, a one-dimensional spectrum of the seismic profile is obtained as an intermediate result, which creates specific conveniences for interpretation of the seismic wave field. The inconvenience of this method, like the Van der Luchta, is the two-step process of producing the autocorrelogram.

Initial experiments devoted to investigating the possibilities of using the optical medium to record a layer of Ag-As_2S_3 semiconducting metal produced by evaporation without physicochemical processing were initially conducted to eliminate the inconveniences of the two-step process. The experiment was conducted by the scheme shown in Figure 5, where 1 is a laser, 2 is a semitransparent mirror, 3 and 4 are a filter-collimating unit, 5 is microscopy of the seismic profile, 6 and 7 are an astigmatic lens, 8 is the medium and 9 is a mirror. A wavelength of = 514.5 nm was used for holographic recording while the carrier frequency of recording was equal to 1,000 lines/mm. The spectra are multiplied after the reference wave has been eliminated and an autocorrelogram of the input object is observed in the output plane [3].

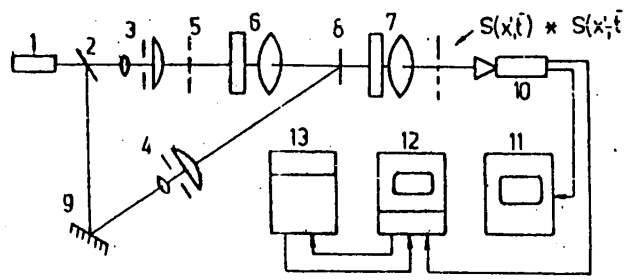


Figure 5. Optical-Electronic System of Multichannel Holographic Correlator: 1--laser; 2--semitransparent mirror; 3 and 4--filter-collimating unit; 5--microscopy of seismic profile; 6 and 7--astigmatic lens; 8--medium; 9--mirror; 10--television camera; 11--monitor; 12--storage oscilloscope; 13--microcomputer

Overcoming the two-step process of finding the autocorrelogram and autocorrelation functions of the seismic profile creates real prerequisites for developing a more complex multichannel optical-electronic device which can be included in the total cycle of machine processing of seismic information: in an optical spectral analyzer (OSA)--filtration by frequency, by apparent velocities (indirection) and by wave number, and in a correlator--finding the autocorrelogram and autocorrelation functions. This system permits one to realize different combinations of filtration parameters with relatively low expenditures of time and funds and estimation of the effectiveness of these parameters. One has in mind in this case that digital filtration of such a large volume of information occupies much machine time, while optimization of the operator for filtration makes processing more expensive as a whole.

The purpose of making a spectral-correlation analysis of the results of "24-fold addition" is to estimate the effectiveness of suppressing multiple seismic waves which make difficult clear geophysical interpretation of the investigated profile. The derived autocorrelogram shows the presence of intensive multiple waves with a delay with respect to the multiple-generatrix in the range $\tau = 0.02-0.2$ second. Autocorrelation functions (Figure 4) which characterize the presence of different types of reverberation systems were found on some segments of the autocorrelogram. Accurate and complete interpretation of the results requires a knowledge of additional geophysical and geological information and is the object of other investigations. The effectiveness of attenuation of multiple waves by "24-fold addition" (coefficient K) was estimated and the unified statistical characteristic of factors was found that determine the possibility of separating single waves from interfering multiple waves (coefficient P):

$$K = \frac{A_i}{A_0},$$

where A_i is the amplitude of auxiliary maximums and A_0 is the amplitude of the main maximum;

$$P = \frac{\tau_1 \tau_2}{m A \tau_3},$$

where τ_1 is the distance along the time axis between the first lateral maximum and the main maximum, τ_2 is the distance between the main maximum and the lateral maximum of the greatest amplitude, τ_3 is the width of the main maximum, m is the number of auxiliary maximums and A is the total amplitude.

Data for coefficients K and P are presented in Table 1. The results show that "24-fold addition" is sufficiently effective for sections II and III (Figure 4), but it is not sufficiently effective for some multiple waves ($K > 0.1$). It is obvious from the given values of P that favorable sections for separation of single waves are III and V ($0.9 < P \leq 2$), while unfavorable sections are all the remaining sections ($P < 0.9$).

The results of the experiments, like their geophysical interpretation, show the possibility of including the described analysis in the entire cycle of processing and interpretation of seismic information. Development of an optical-electronic spectral analyzer-correlator, which will simultaneously accomplish several processing algorithms, is especially promising. This device may also find application in

Table 1.

	$K = A_i/A_0$					
i	I	II	III	IV	V	VI
1.	0,192	0,075	0,086	0,206	0,168	0,152
2.	0,048	0,083	0,021	0,206	0,105	0,086
3.	0,064			0,049	0,032	0,057
4.	0,024			0,039		0,067
5.				0,049		0,200
6.				0,020		0,038
7.				0,029		0,029
8.				0,020		0,019
9.						0,019
P	0,490	0,490	2,01	0,070	1,08	0,230

in other multichannel geophysical investigations (for example, aeromagnetic investigations and aerogravitational investigations) which are related to processing and interpretation of large files of geophysical information.

BIBLIOGRAPHY

1. Potapov, O. A., "Opticheskaya obrabotka geofizicheskoy i geologicheskoy informatsii" [Optical Processing of Geophysical and Geological Information], Moscow, Nedra, 1977.

2. Fontanell, A. and G. Grau, "Correlation Optique en Lumiere Coherente," GEOPHYSICAL PROSPECTING, No 1, 1971

3. Goodman, J., "Vvedeniye v fur'ye-optiku" [Introduction to Fourier Optics], Moscow, Mir, 1970.

6521
CSO: 1863/26

UDC 535

HOLOGRAPHIC METHOD OF CHECKING REFLECTORS

Leningrad GOLOGRAFIYA I OPTICHESKAYA OBRABOTKA INFORMATSII: METODY I APPARATURA in Russian 1980 (signed to press 31 Dec 80) pp 229-230

[Article by A. P. Golikov, M. L. Gurari and S. I. Prytkov from the collection "Holography and Optical Processing of Information: Methods and Apparatus", edited by V. G. Skrotskiy, B. G. Turukhano and N. Turukhano, Izdatel'stvo Leningradskogo instituta yadernoy fiziki, 500 copies, 237 pages]

[Text] The development of a new holographic method to measure the radii of curvature and local surface distortions of large mirrors are reported [1].

Serial production of large-diameter concave mirrors and especially of convex mirrors is now weakly supported with checking equipment. The complexity of developing nonaberration wave fronts of large aperture leads to an increase in the errors of interference methods [2]. Methods that do not require special reference components --shift interferometry and interferometry with a scatterer--are applicable to analysis of only convergent radiation beams. It is significant for these methods that the base of the measuring installation increases in proportion to the radius of curvature of the mirror being checked, which in the case of concave mirrors of small curvature makes the installation cumbersome and leads to a reduction of measurement accuracy due to an increase of interference and in the case of convex mirrors requires the use of large-size compensators.

A schematic diagram of the experimental installation is presented in Figure 1. The mirror to be investigated is illuminated by a diverging flux of LG-38 laser emission through a scatterer arranged tightly against the mirror. The reflected flux through this scatterer impinges on a photographic plate and its hologram through which the surface of the scatterer is subsequently observed is recorded on it. A shift of the mirror leads to formation of an interferogram localized on the scatterer. The geometric parameters of the surface of the mirror being checked can be determined by the shape of the interferogram.

An increase of measurement accuracy is achieved by introducing compensatory rotation of the mirror in addition to shifting of it. The interferogram is reduced to standard form corresponding to rotation of the mirror around its center of curvature by the combination of shift and rotation. The radius of curvature is found by the measured values of shift and rotation, while the distortion of the mirror surface is found by the shape of the bands of the interferogram.

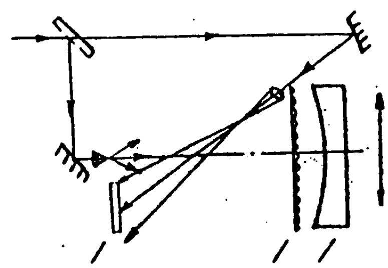


Figure 1. Schematic Diagram of Experimental Installation

The holographic measuring installation with scatterer installed in front of the mirror can measure the radii of curvature of mirrors from 200 mm and above in diameter with an accuracy on the order of 0.1 percent (for radii of curvature in the range from 10 to 50 meters). The overall dimensions of the installation over a wide range are not dependent on the curvature of the mirrors being checked and precision optics is not used during measurement. The relatively small base of the measuring device and its independence of the curvature of the surface being checked permits one to use the suggested method for production checking of metal-optics under plant conditions.

BIBLIOGRAPHY

1. "USSR Inventor's Certificate No 593070," BYULLETEN IZOBRETENIYE, No 6, 1978.

2. Dukhopel, I. I. and L. G. Fedina, OPTIKO-MEKHANICHESKAYA PROMYSHLENNOST', No 7, 1973

6521
CSO: 1863/26

UDC 535

BEAM HOLOGRAPHY

Leningrad GOLOGRAFIYA I OPTICHESKAYA OBRABOTKA INFORMATSII: METODY I APPARATURA in Russian 1980 (signed to press 31 Dec 80) pp 231-234

[Article by N. A. Valyus from the collection "Holography and Optical Processing of Information: Methods and Apparatus", edited by V. G. Skrotskiy, B. G. Turukhano and N. Turukhano, Izdatel'stvo Leningradskogo instituta yadernoy fiziki, 500 copies, 237 pages]

[Text] An analog of the holographic image obtained in coherent light is an integral photograph recorded in noncoherent light.

An integral photograph, like holography, is characterized by "complete recording" about the object, its volume and its relative position in space. The same pattern can be recreated from the part of an integral photograph and from a fragment of it as from the entire photograph.

The image of an integral photograph may be called beam holography since only the amplitudes rather than the phase of the wave are recorded when using noncoherent light and a description of the process is found within the framework of geometric (beam) optics.

The principle of integral photography was suggested at the beginning of this century by G. Lippman, more well known for his idea of interference photography, which was a prototype of the modern reflective hologram substantiated and developed by Yu. N. Denisyuk.

G. Lippman [1] called his method integral photography because the three-dimensional image produced is synthesized from images of a large number of small microscopic images of the object photographed with a scanning lens photographic plate and one can say the restored wave front is integrated from the number of elementary beams represented by these images.

The process of photographing a three-dimensional image on a scanning lens plate is shown schematically in Figure 1. One surface of the scanning lens plate R has fluting in the form of small lenses contacting each other, while the other surface is coated with a light-sensitive layer on which the small lenses are focused. The object AB, located in front of the lens side of the plate, is exposed on the light-sensitive layer by the small lenses in the form of a set of microscopic images a_1b_1, a_2b_2, a_3b_3 ...

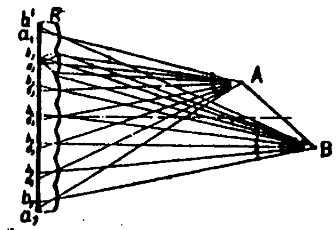


Figure 1.

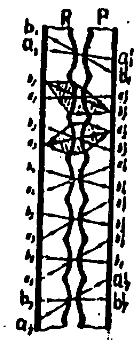


Figure 2.

A photograph can be made on this plate without the use of a lens.

The process of achieving an integral photograph is a two-step one. Therefore, after the plate R has been exposed, developed and fixed, it is placed opposite the other, exactly the same plate P, turning the plates with the lens elements toward each other (Figure 2) and they are illuminated from behind by scattered light. The inverted positive image is copied, i.e., elementary images a_1b_1, a_2b_2, ... are inverted to images $a'_1b'_1$, $a'_2b'_2$, ...

The second plate is a positive image which restores the inverted path of the beams from the object (the quantified wave front from the points of the object is restored). By illuminating this plate P on the emulsion side and looking at it from the lens side (Figure 3), one can see a three-dimensional image of the photographed subject AB'.

The three-dimensional image of the subject is recreated by the wave fronts in the directions of beams from each elementary image. Therefore, the visible image is imaginary and is located at a quite specific distance on the plate.

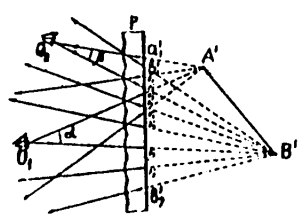


Figure 3.

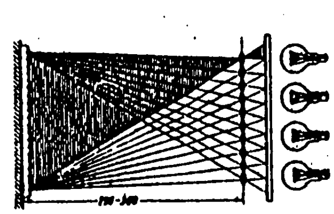


Figure 4.

One can see a side view of the object when viewing the integral photograph from the side. When one approaches the panorama of the visible picture is expanded and the perspective and angular size of the subjects change.

An integral photograph made in noncoherent light permits filming of full-scale three-dimensional objects of unlimited dimensions at arbitrary distances and permits color images to be made.

Different modifications of integral photography are known [2]. Thus, for example, it is possible to increase the three-dimensional resolution of the produced image by constriction of the aperture when taking an integral photograph through a large-diameter objective (or lens). A synthesized integral photograph can be made when the full-scale object is photographed on a set of frames from different angles of approach and then the print of these frames is projected from these same angles of approach onto a scanning lens plate, as is shown in Figure 4. Conversion of a focused integral photograph by coherent radiation to a reflective hologram with altered three-dimensional image scale yields interesting results.

Quasi-integral photographic systems, so-called autostereoscopic images, which find application in the form of three-dimensional advertising images and as mass stereoscopic isoproduction, are of practical interest.

The principles of beam holography accomplished by using scanning optical systems permit one to solve many problems not accessible by traditional classical devices.

Scanning integral stereoscreens, scanning objectives and eyepieces are known. Scanning optical systems provide recording not only of the three-dimensional angles of approach of an object but of its time angles of approach as well. The highest movie filming speeds are achieved by using scanning frame cameras. Scanning optical systems are promising in computer equipment as storage devices, as processors, controllable slides and so on [3].

Finally, one should also point out the analogy of scanning systems to holographic systems in the field of interferometry. Essentially the Moire phenomenon is interference of three-dimensional frequencies of raster gratings that is used in measuring equipment.

BIBLIOGRAPHY

1. Lippman, G., JOURNAL DE PHYSIQUE, 1908.

2. Valyus, N. A., "Rastrovaya optika" [Scanning Optics], Moscow-Leningrad, GITTL, 1949.

3. Valyus, N. A., "Matrix Type of Optical Computers," "Radioelektronika opticheskogo diapazona" [The Electronics of the Optical Band], Moscow, VZMI, 1970.

6521
CSO: 1863/26

SOFTWARE

ABSTRACTS OF ARTICLES IN JOURNAL 'PROGRAMMING', SEPTEMBER-OCTOBER 1981

Moscow PROGRAMMIROVANIYE in Russian No 5, Sep-Oct 81 pp 95-96

UDC 51 : 621.391

PROBLEM OF EQUIVALENCE FOR GENERALIZATION OF LL(k) GRAMMARS

[Abstract of article by Nepomnyashchaya, A. Sh.]

[Text] A proof is given of the solvability of the problem of equivalence for grammars of P_m^k classes, which are specified by uniting mLL(k) grammars with limitations on outputs. P_m^k classes, as well as deterministic automata with a magazine memory corresponding to them, represent new classes with a solvable equivalence problem.

UDC 519.767.2

DESCRIPTION OF SEMANTICS OF PROGRAMS BY MEANS OF SUBSTITUTIONS IN GRAPHS

[Abstract of article by Gostev, Yu.G.]

[Text] It is suggested that programs be interpreted as combinations of substitutions to be applied to graphs of a certain kind. This makes it possible to draw the theory of algorithms closer to methods of describing computations used at the present time and in addition makes it possible to describe computations on random data structures.

UDC 681.3.06

SEMANTIC MODELS OF PROCEDURES

[Abstract of article by Pastor, F.]

[Text] Semantic models of procedures and other objects encountered in programs written in a high-level programming language are discussed. A method is suggested for constructing complicated models of statements from simpler ones, based on a composite approach.

UDC 681.3.06

PROGRAMMING SUBSTITUTIONS OF LANGUAGES

[Abstract of article by Cheremisinov, D.I.]

[Text] The problem of programming substitutions of languages is discussed--the operation of converting string data forming the basis for constructing different kinds of algorithms for processing data represented by chains of characters. A system of notation for defining substitutions of languages is suggested.

UDC 681.3.06

CONSTRUCTION OF A DATA BASE BASED ON THE CONCEPT OF ABSTRACT TYPES

[Abstract of article by Zamulin, A.V. and Skopin, I.N.]

[Text] The concept of astract types of data as applied to designing systems for controlling data bases is discussed. The procedure discussed makes it possible on the basis of a universal programming language with abstract types to develop data bases in accordance with a specific application model.

UDC 681.142

SOFTWARE IMPLEMENTATION OF MULTIPROCESSING

[Abstract of article by Godunov, A.N., Yemel'yanov, N.Ye. and Sverdlov, S.S.]

[Text] A description is given of a multiprocessing executive routine (MP executive) created within the framework of the INES-2M [economic information system] SUBD [system for control of data bases], which can also be used independently. The MP executive makes possible parallel execution of the computing process and working with a disk-type virtual memory utilizing the distinctive features of the multiprocessing mode of executing a routine.

UDC 681.3.06

IMPLEMENTATION OF A 'FOREKS' COMPILER FOR THE AS-6 CENTRAL PROCESSOR

[Abstract of article by Galatenko, V.A. and Khodulev, A.B.]

[Text] A description is given of the instruction generator in the compiler from FORTRAN for the central processor of the AS-6 complex. Information is presented on organization of the AS-6 central processor. The characteristics of the AS-6 central processor and BESM-6 are compared. An evaluation of the AS-6 central processor as a FORTRAN machine is given.

UDC 681.3.06

INFORMATION SEARCH AS A PROBLEM IN THE LINGUISTIC RECOGNITION OF CHARACTERS

[Abstract of article by Samedova, M.A.]

[Text] The linguistic support of a data search of objects of a medical nature represented in a normalized natural language is discussed.

UDC 681.3

SYSTEM OF DIALOGUE PREPARATION OF TASKS FOR UNIFIED-SERIES COMPUTERS

[Abstract of article by Khanykov, V.V., Rybakov, A.V. and Anan'ina, N.V.]

[Text] A description is given of an instrument system for working with a design library for YeS computers within the framework of the operating system.

UDC 681.3.06 : 51

EVALUATION OF INDICATORS OF THE CORRECTNESS OF THE FUNCTIONING OF PROGRAMS WITH A MULTILEVEL STRUCTURE

[Abstract of article by Khaletskiy, A.K.]

[Text] A mathematical formulation is suggested for the problem of evaluating the vector of indicators of the correct functioning of a complicated program with a tree-type hierarchical structure fpr the case when failures of its components are independent.

UDC 681.3.06

RECURSIVE ESTIMATION OF ARITHMETIC FUNCTIONS IN LISP SYSTEMS

[Abstract of article by Stefanyuk, V.L.]

[Text] In this note reasons are presented in favor of the "explicit" computation of elementary functions in LISP systems. Examples are given of simple recursive codes for computing sin , exp and ln , illustrating the general approach.

UDC 681.3.001

COMBINED FORMAT FOR OUTPUT OF NUMERICAL INFORMATION FOR DESIGN DOCUMENTS

[Abstract of article by Motyl', D.N., Sokolinskiy, Yu.A. and Farber, K.S.]

[Text] A description is given of a combined format for the representation of numerical information making it possible to output numbers with a specified relative error. The described format is implemented in the procedure function of the PL/1 language in the YeS [Unified Series] operating system.

CSO: 1863/45

UDC 681.142

SOFTWARE IMPLEMENTATION OF MULTIPROCESSING

Moscow PROGRAMMIROVANIYE in Russian No 5, Sep-Oct 81 (manuscript received 12 May 80) pp 44-49

[Article by A.N. Godunov, N.Ye. Yemel'yanov and S.S. Sverdlov]

[Text] A description is given of a multiprocessing executive routine (MP executive) created within the framework of the INES-2M [economic information system] SUBD [system for control of data bases], which can also be used independently. The MP executive makes possible parallel execution of the computing process and working with a disk virtual memory utilizing the distinctive features of the multiprocessing mode of executing a routine.

As mentioned in [1, 2], the idea of multiprocessing, i.e., breaking down a user's task into independent quasi-parallel processes, is very attractive both from the viewpoint of the convenience of programming and on the plane of possible optimization of working with an external storage. Furthermore, it was indicated that the most advantageous means of parallel execution would be the hardware implementation of a multiprocessing system. Nevertheless, a considerable gain can be made also with software implementation of such a system.

In this article a description is given of a real multiprocessor executive system which can be used both for creating individual user routines and as a system for programming various components of the software of an ASU [automated control system], data base control systems and the like. Although the MP executive has been implemented within the framework of the INES SUBD [3], it represents a totally independent system which can be used independently or as part of any software complex under the control of a YeS [Unified Series] operating system.

The following were the key requirements in creation of the MP executive:

Offering the user a developed system of macroinstructions for parallel execution of the computing process.

The creation of convenient facilities for working with a virtual memory employing disks.

Ensuring the effective control of swapping in the multiprocessing mode.

Making possible the auxiliary functions required for practical work with the executive routine (debugging facilities, message output facilities, management of routine execution protocols and, in particular, swapping procedures, and statistics gathering facilities).

Ensuring compatibility of the MP executive with the YeS operating system.

The functional capabilities of the MP executive are illustrated in fig 1, where 1 represents facilities for working with the virtual memory, 2 facilities for parallel execution and the synchronization of processes, 3 debugging facilities and 4 facilities for organizing parallel message flows.

Multiprocessing Instruction Set

The parallel exeuction of a routine which is to be executed in the multiprocessing mode takes place through special macroinstructions. The following macroinstructions exist for the origin, termination and synchronization of processes: INIT--initiation of multiprocessing operation, PROCS--origin of a process, PREND--termination of the process, HALS--cancellation of all processes begun, PRTY--changing the priority of a running process, MAYINT--enabling suspension of a running process, WALS--awaiting the termination of all processes started, WANS--awaiting the termination of any process started, SM--awaiting the assigned value in the semaphore and WAITM--awaiting an operating event (in the sense of the operating system).

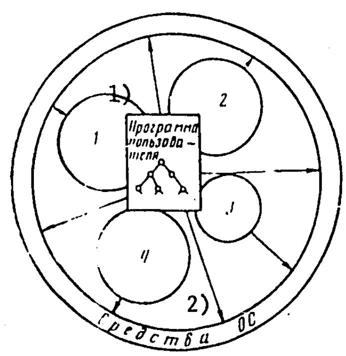


Figure 1.

Key:
1. User's program
2. Operating system facilities

In the initiation of a multiprocessing system its initial configuration and starting parameters must be specified, such as the maximum number of processes existing simultaneously in the system, the maximum depth of the process origin tree, instructions for use of the virtual memory, a program identifier for the initial process, some additional instructions for the accidental termination of processes, debugging printout and the like.

During initiation the system's control tables are formed, special system processes are created and the initial user's process is started.

Any process can issue a PROCS instruction for the origin of an offspring process. If there are insufficient resources for the creation of a new process, the parent process is halted until the necessary memory is freed. After the creation of the process it, as a rule, also starts, since, if nothing special is stipulated, the offspring process receives higher priority than the parent process. In such a manner the multiprocessing system stimulates growth of the process tree "in depth," and not "in breadth," in order to reach more rapidly the maximum depth at which processes begin to end, issuing the instruction PREND. The memory freed at the termination of processes is used for producing new processes. There is also the possibility of arbitrarily assigning priority to a begun process and, in addition, each process in the course of execution can change its priority by means of the PRTY instruction. This can result in dead-end situations when all processes begun wait for resources for the creation of offspring processes. If the assignment of priorities is left to the discretion of the MP executive itself, then it provides certain measures for the prevention of dead-end situations, on the basis of the maximum depth of the process origin tree indicated during initiation.

The parent process can cancel all processes originated by it by means of the HALS instruction, whereby each offspring process is cancelled together with all processes subordinate to it. The WALS instruction suspends the running process until the end of the operation of all offspring processes produced by it, and the WANS instruction until the end of any offspring process (if there are any). The most common means of synchronizing processes is the semaphore device, whereby the semaphore can be any memory cell. By means of the SM instruction an indication can be given regarding the entry of a certain value into the semaphore, regarding waiting for a specific value or condition in the semaphore, and regarding waiting followed by an entry into the semaphore.

Virtual Memory of a Multiprocessing System

The multiprocessing instruction set includes the following facilities for working with a virtual memory employing disks: OPENM--open the file of the virtual memory, CLOSEM--close the file of the virtual memory, EXFIL--designate the working file of the virtual memory, LV--load into the register the word with the indicated virtual address, STV--store contents of the register according to indicated virtual address, MVV--move field from virtual memory to virtual memory, MVS--move field from ordinary memory to virtual, MSV--move field from virtual memory to ordinary, CVV--compare field of virtual memory, CVS--compare field of ordinary memory and virtual memory, INPG--input page with indicated virtual address, GETPG--issue clean page with indicated virtual address, UPDPG--set page correction tag, FREPG--free page with virtual address indicated, FIX--make request for fixing of the required number of virtual memory pages.

The MP executive makes it possible to work simultaneously with several virtual files. Each process must open the file needed by it by means of the OPENM instruction and upon the end of working with it must close it by means of the CLOSEM instruction. Actual closing of the file in the sense of the YeS operating system is produced by the MP executive when the last process opening it finishes working with this file. In addition, each process is assigned, if this is necessary, an individual working file of an indicated size, it is opened and closed automatically by the MP executive, and accessing it does not require identification of the file.

The LV and STV instructions are intended for working with words of the virtual memory and are similar to the assembler L and ST instructions. The MVV, MVS, MSV and CVV and CVS instructions are similar to the assembler MVC and CLC instructions, respectively, but the length of operands can reach 32,768 bytes. The INPG instruction makes it possible to gain access to a page with the indicated virtual address and the requested page will be fixed in the working storage right up to a clear indication by the programmer of completion of work with it by means of the FREPG instruction. The UPDPG instruction is used in order to indicate which page entered by means of the INPG instruction will be corrected and, consequently, later must be automatically output according to the corresponding virtual address. By means of the GETPG instruction it is possible to obtain any "clean" page, i.e., a working storage buffer. The correction tag for this page is set automatically, in order for it later to be read out into the appropriate virtual file. Let us note that in execution of the MVV, MVS and STV instructions the page correction tag is also set without an explicit instruction from the programmer.

As in indicated in [1, 2], multiprocessing opens up unique opportunities for considerably increasing the effectiveness of a virtual memory. The simultaneous existence of a great number of processes, each of which possibly issues requests for exchange with virtual files makes it possible to implement an effective exchange algorithm which stimulates the partial review in advance of a sequence of accesses to the virtual memory [2]. There also is discussed an algorithm for protection from dead-end situations in a multiprocessing system which can originate when working with the pages of a virtual memory under fixing conditions.

Additional Possibilities Offered by the MP Executive

The existence in the system of a great number of independent processes creates serious difficulties in organizing the printout of the output messages of each of them. In quasi-parallel running of a user's program the sequence of the execution of individual processes is determined dynamically by the multiprocessing system and, consequently, the procedure for the output of messages by various processes is unpredictable. It is obvious, however, that from the viewpoint of operating convenience the separation of flows of messages of various processes in printout is totally necessary. The use of standard YeS operating system facilities for these purposes is exceedingly difficult for an applied programmer because of their cumbersome nature. The use of the printout facilities offered by the MP executive makes possible the automatic separation of flows of messages of various processes and in addition frees the programmer from organizing control tables of data (DCB's) and from the need to perform the operations of opening and closing data sets. Here the re-enterability of the routine executing the printout is not disturbed.

With the accidental termination of some process the MP executive makes possible the output of a standard diganostic message regarding the type of error, a list output of the area of interruption and of the state of registers at the moment of interruption, and also a list output of key multiprocessing control tables at the moment of interruption.

In the debugging mode the MP executive makes it possible at points indicated by the programmer to print out the current state of control tables of the multiprocessing system and of process waiting lines. When working with files of the virtual memory there is the possibility of periodically (at a given interval) obtaining information on exchanges carried out and on the current state of virtual memory buffers. Information on exchange dynamics in the multiprocessing system can be very helpful for a qualitative analysis in selecting optimum multiprocessing parameters for a specific problem (number of processes, number of buffers, virtual memory, their size and the like).

Functioning of a Multiprocessing System

The execution of multiprocessing instructions is performed by program modules included in the MP executive's structure. The completion of an instruction by the MP executive can involve either continuation of a running process or switching to another process. For example, if in a reference to the virtual memory it is found that the page indicated is already in the working storage, the running process again receives control. Otherwise it is suspended with the appropriate waiting code, its request for a page of the virtual memory remains in the waiting line for execution and control is transferred to one of the processes ready to run.

The MP executive maintains waiting lines of processes waiting for a specific event and the waiting line of so-called ready-to-run processes, from which the process with the highest priority is selected in switching. The state of these waiting lines is constantly corrected with the completion of each multiprocessing instruction by the executive routine. For example, execution of the PREND instruction for a certain process can result in placement of the parent process in the ready-to-run waiting line (if it was awaiting termination of the running of all offspring processes and the running process was the only one or the last of them), as well as one of the processes awaiting memory resources for the creation of an offspring process.

Special system process W performs the function of waiting for any event (in terms of the operating system) which is specified by the process. Among them, process W fixes the end of any exchange with external units. Let us note that all processes, in addition to facilities for working with the virtual memory, can order exchange with external units through ordinary machine instructions of the operating system. Process W has minimum priority and thus receives control when it remains the only one of the processes ready to run. Furthermore, process W reviews the waiting line of events and transfers the individual processes into the ready-to-run state. If not one of the awaited events takes place, process W issues the operating instruction WAIT.

System process E implements the algorithm for exchange with the virtual memory described in [2]. Its functions include review of the waiting line of requests for exchange with the virtual memory and initiation of the exchange selected. Process E has maximum priority in the system, which makes it possible to insure most complete matching of the operation of the processor and input/output channel.

System processes W and E are created by the MP executive upon the initiation of multiprocessing activity, and process E is generated only when an instruction regarding working with the virtual memory is contained in the INIT macroinstruction.

As is obvious from a description of multiprocessing instructions (cf. fig 2), some of them are similar to operating macroinstructions of the YeS operating system associated with a multitask situation. Let us note that each process of the MP executive is at the same time a subtask in terms of the operating system. However, the issuing of ATTACH operating macroinstructions is necessary only upon the initiation of a multiprocessing operation and in cases of the accidental termination of certain processes. With the normal termination of a process and the creation of a new one, the issuance of DETACH and ATTACH macroinstructions is not required, since one and the same operating system subtask is used many times in succession for supporting a great number of MP executive processes. Execution of the HALS macroinstruction takes place similary without accessing the operating system. Instructions for the synchronization of processes on the one hand give a programmer more convenient facilities than operating macroinstructions. For example, often the common function of waiting for all subordinate processes is implemented in the MP executive as a single simple WALS instruction. On the other hand, instructions for working with semaphores, although they make it possible to simulate the operating facilities for working with ENQ and DEQ resources, are completed by the MP executive without using these system macroinstructions, which makes it possible to speed up execution of the program.

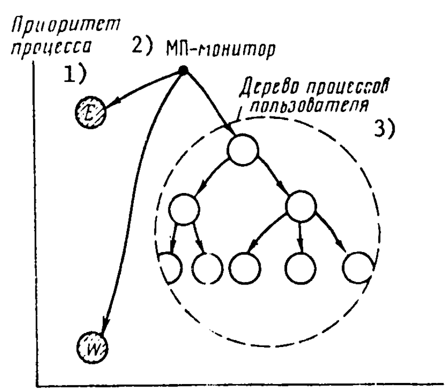


Figure 2.

Key:
1. Priority of process
2. MP executive
3. User process tree

Let us stress that although each process of the MP executive is also a subtask of the operating system the function of the process scheduler is performed exclusively by the MP executive. With each accessing of the executive routine it selects for execution a single process (subtask of the operating system) and the remaining are inhibited, even if they are ready to continue running. At the same time in the interval between two accesses to the MP executive each process is uninterruptable in a given multiprocessing mixture. This opens up the possibility of extensively using quasi-re-enterable routines for processes, the creation of which is less difficult for a programmer than ensuring their absolute re-enterability. (The maintenance of re-enterability of some sort is necessary in the widespread situation when a great number of processes are designed to run according to a single routine.)

The concentration of the functions of controlling processes in the MP executive makes it possible to achieve good agreement between the work of the central processor and the execution of input/output operations. For this it is sufficient to use for the purpose of waiting for the end of exchanges the multiprocessing macroinstruction WAITM, which results in suspension of the running process and the start of another process ready to run. The possibility of organizing exchanges of just operating system facilities is also not excluded (without accessing the MP executive), which can entail only a slight reduction in the effectiveness of the system.

Moreover, the job of controlling the multiprocessing system does not contradict the use in programs of standard broad-application operating system macroinstructions, such as GETMAIN, LINK, LOAD and the like, the use of which can be dictated by the programmer's support considerations. This compatibility of multiprocessing with operating system facilities makes it possible also to adapt fairly easily to a multiprocessing mode a program written by the traditional sequential method. For this is required mainly a "superstructure" of the program's upper logic level making possible its parallel execution.

The methods of multiprocessing discussed can prove to be a convenient tool for both system and applied programming. The instruction set described above provides the programmer with added conveniences as compared with the multitask facilities offered by the YeS operating system and furthermore is implemented in order to reduce as much as possible the number of expensive accesses to the operating system's supervisory routine. The convenience of programming in terms of multiprocessing was regarded as an important factor in creating the MP executive. Therefore the executive routine has a structure making it possible easily to add additional instructions for specific applications and to adapt the multiprocessing system to the user's needs.

The functional capabilities of the MP executive make it possible to use it in a broad class of data processing tasks [1, 2] and in software design system tasks. On the basis of the MP executive a multiterminal SUBD INES system has been designed which is described easily and naturally in the language of parallel processes.

Bibliography

1. Arlazarov, V.L., Volkov, A.F., Godunov, A.N., Yemel'yanov, N.Ye., Zenkin, V.D., Konstantinov, G. and Lysikov, V.T. "Hardware Implementation of Multiprocessing," AVTOMATIKA I TELEMEKHANIKA, No 8, 1977.

2. Godunov, A.N. and Sverdlov, S.S. "Organization of a Virtual Memory in a Multiprocessing System," AVTOMATIKA I TELEMEKHANIKA, No 12, 1978.

3. Arlazarov, V.L., Dyukalov, A.N., Yemel'yanov, N.Ye., Ivanov, Yu.N., Kochin, Yu.Ya., Tokarev, V.V. and Faradzhev, I.A. "The INES Information System," AVTOMATIKA I TELEMEKHANIKA, No 6, 1979.

8831
CSO: 1863/45

UDC 681.3.06.

IMPLEMENTATION OF 'FOREKS' COMPILER FOR AS-6 CENTRAL PROCESSOR

Moscow PROGRAMMIROVANIYE in Russian No 5, Sep-Oct 81 (manuscript received 24 Dec 80) pp 50-58

[Article by V.A. Galatenko and A.B. Khodulev]

[Text] A description is given of the implementation of a FOREKS FORTRAN compiler for the central processor of the AS-6 complex. A brief description is given of the AS-6 central processor. The results of statistical studies of object programs are given.

The AS-6 is a multimachine complex which can include central processors (AS-6 TsP's), a BESM-6 computer and peripheral machines (PM-6's). The architecture of the AS-6 central processor is rather complicated. Let us mention, for example, the presence of three types of registers, the page segment organization of the memory, and the extensive and quite individual instruction set.

In this article a description is given of a compiler from FORTRAN for the AS-6 TsP obtained by modifying the FOREKS compiler for the BESM-6 [1, 2]. The FOREKS is a two-operation compiler. In the first operation the program is translated from FORTRAN to the internal language and in the second from the internal language into machine instructions. The internal representation used by the compiler for the BESM-6 proved to be totally suitable for the AS-6 central processor; therefore, only the instruction generator is a part of the compiler specific to the AS-6 central processor. It is precisely to the instruction generator for the AS-6 central processor that this article is mainly devoted. In addition, in this article data are presented which characterize the AS-6 central processor as a FORTRAN machine.

The instruction generator for the AS-6 central processor, as are the other parts of the FOREKS compiler, is written in the ASTRA language [3] and operates in the BESM-6, producing a text in AS-6 central processor autocode [4]. The capacity of the generator is about 3500 ASTRA instructions (about 6000 BESM-6 words).

The input language of the FOREKS compiler conforms basically to the FORTRAN-77 standard [5] and also contains facilities absent in FORTRAN-IV, such as "structural" conditional instructions, IF-THEN-ELSE-ENDIF, cycle instructions, DO-REPEAT (in combination with the EXIT statement), as well as the CHARACTER (character change) and BIT (bit change) data types.

1. Brief Description of AS-6 Central Processor

In this section are described only those features of the AS-6 central processor which influence the instruction generator. More detailed information on the AS-6 central processor, including a description of the instruction set, can be found in [4].

1.1. Addressing Mechanism

The memory in the AS-6 central processor is virtual and segmented. The maximum size of a segment equals 2^{18} 48-bit words. One task can work with no more than 4096 segments. The virtual memory is accessed by means of 48-bit descriptor registers. There are 16 of these registers. They contain the following information:

1. The number of the segment which is accessed (12 bits).

2. The address within the segment. This address is called the base and occupies 24 bits.

3. The type of base (2 bits). In the AS-6 central processor it is possible to address not only whole words, but also half-words, bytes and bits. The type of base shows in which kind of units the address within a segment is given.

4. The number of the index register or half-word adder through which indexing will be performed.

The descriptors contain also other information which is not important for our purposes. There are eight 24-bit index registers in the AS-6 central processor in addition to the descriptor registers. The operations which can be performed in them are addition and subtraction but their main function is the modification of addresses. In an AS-6 central processor instruction the address of an operand in the memory is given by means of the number of the descriptor register, the shift (16 bits) and the indexing tag. The operand is accessed from the segment indicated in the descriptor register. Its address within a segment is computed as the sum of the shift indicated in the instruction, of the base of the descriptor and, if an indexing tag has been set, the contents of the index register or half-word adder whose number is given in the descriptor. The type of descriptor base must agree with the type of operand or be a word base. In the latter instance before being added with the shift and contents of the index register or half-word adder the descriptor's base is multiplied by 2, 6 or 48.

Descriptor registers can be used not only for addressing, but also for organizing cycles: In the descriptor register are placed half-word steps and the final value of the control variable; the instantaneous value of the control variable must be stored in the index register. There is a 3-operand cycle end instruction which with a positive value of the step is similar to the BXLE instruction in YeS [Unified Series] computers (cf. [6]). If the step is negative, a transfer is made, when the new value of the control variable is not lower than the final value.

2. Adder Registers

The eight 48-bit adder registers are designed for performing arithmetic operations with a floating and fixed point, as well as logic operations.

Numbers with a floating point have a 40-bit mantissa, a 7-bit floating point part and a sign bit. Negative numbers are represented in direct code. The floating point portion is hexadecimal and normalization of the fixed point portion is performed with an accuracy of a hexadecimal digit. The range of representable numbers is from 16^{-65} to 16^{63}.

The set of operations on numbers with a floating point is the traditional one. The mode for the execution of arithmetic operations is indicated in a special interlocking register. Normalization of the result can be interlocked, as well as overflow and loss of significance interrupts, etc.

In the AS-6 central processor there are also instructions for working with double-precision numbers. These numbers can be added, subtracted and multiplied. There is no instruction for dividing numbers with double precision.

Whole numbers can be represented by two methods. Word whole numbers contain 47 bits and a sign. Ordinary arithmetic and logic operations, in addition to the division operation, can be performed on these numbers.

Half-word whole numbers contain 23 bits and a sign. The right halves of adders, which are called half-word adders, are used when working with them. Negative numbers are represented in complement. The operation set includes arithemetic (including division) as well as logic operations. In addition, half-word adders, as the index registers, too, can be used for modifying addresses.

The AS-6 central processor permits working with individual bytes and bits. Here the far right bits of the adders are used as bit adders. All traditional logic functions are included in the operation set. Operations with bytes are performed in the right byte of adders.

The adder registers can be used for operations with sequences of bytes and bits. Here the length of a sequence must not exceed 12 bytes (96 bits).

3. Memory-Memory Format Instructions

Sequences of bytes and bits can be processed not only in registers but also in the memory. The set of corresponding AS-6 central processor instructions is similar to the instruction set of the SS format in YeS computers with the only difference that the length of operands is not obligatorily specified directly in an instruction: It can be placed in an index register and its number can be indicated in the instruction. Here the length of operands can reach 2^{18} words.

4. Dynamic Loading

The dynamic loading strategy (cf. [7], sec 4.9.2) is used in the AS-6 central processor: Programs are loaded with respect to first access to them. As a result there will be no superfluous programs in the memory. Programs which have been completed and have become unneeded will sooner or later be forced out into the secondary storage.

5. State Stack

A unique feature of the AS-6 central processor is the state stack (hidden stack) mechanism. It makes possible the automatic saving of return addresses and the saving and resetting of registers.

The stage stack operates in the following manner. Accessing of a subroutine is written in the form of two instructions. In the first of them a so-called saving mask is established, indicating which registers are to be saved. The second executes the jump per se. Here in the state stack is registered the return address and certain other characteristics of the module from which accessing takes place. While the subroutine is run the registers indicated in the saving mask are entered into the state stack before the first execution of any instruction altering their contents. Thus, unused registers in the state stack are not stored.

A single return instruction is sufficient in order to exit from a subroutine. It causes the resetting of registers hidden in the state stack and transfers control to the return point.

2. Machine Representation of Language Objects

Numbers with a floating point are represented naturally. It was a more complicated matter to select the representation of whole numbers. We settled our choice on half-word whole numbers in order to render effective access to elements of arrays and the implementation of cycles. Of course, with this the range of representable whole numbers turned out to be rather short: from 8388608 to 8388607.

The structure of the AS-6 central processor makes it possible to address the on-line memory with an accuracy of a bit. Therefore, for logic values the AS-6 FOREKS compiler assigns in terms of a single bit. This makes it possible to work effectively with very large (up to 2^{23} elements) logic arrays.

Thus, regions of the memory of different sizes are assigned to values of different types. This contradicts the FORTRAN-77 standard [5]. However, the memory savings which originates when using whole-number and especially logic arrays in our opinion outweighs this disadvantage.

Values of the CHARACTER and BIT type are represented as sequences of bytes (bits). They are worked with only by means of memory-memory format instructions. Registers are used only in converting from one type to another, e.g., from CHARACTER to INTEGER.

Accessing an element of an array (e.g., A(I)) is a rather complicated operation (setting the descriptor, setting the index and accessing per se). In addition, operations with descriptors are performed rather slowly in the AS-6 central processor. Therefore it is advisable to place arrays as much as possible in a single segment, since then a single descriptor can be used without resetting for working with several arrays. The length of an array cannot exceed the size of a segment (2^{18} words).

An individual segment is assigned to each common (COMMON) block. It follows from this that different common blocks are protected from one another and improper working with one block cannot influence another.

3. Algorithms Used

Methods of translating from FORTRAN are described in detail, e.g., in [8]. We will dwell briefly on some algorithms used in the instruction generator for the AS-6 central processor.

1. Generation of Instructions for Line Sections

A sequence of FORTRAN program instructions which does not contain control jump labels and instructions (a line section), after syntactical analysis and optimization, is represented in the form of a graph of triads (cf. [8]). Each triad contains an operation indicator and references to triads representing operation operands. Triads of a special form represent variables. For a specified graph it is possible to generate various sequences of instructions for its execution differing in the computation procedure for triads and in the use of registers. The following recursive algorithm is used in the realization described for the purpose of generating triad computation instructions.

1. Instructions are generated for computing the second operand of the triad (if they have not already been generated).

2. The same for the first operand.

3. Instructions are generated for the performance of an operation of a triad on its operands.

This algorithm is applied seuqentially to all triads representing instructions of the source program. The use of registers is described below. Although this generation procedure is not always optimum (cf. [9]), practically speaking it makes it possible to produce a sequence of instructions close to optimum.

2. Distribution of Registers

In the AS-6 central processor there are eight adders, eight index registers and 16 descriptor registers. The adders and index registers are distributed uniformly. A table of registers is constructed. Each element of this table contains a reference to the triad which this register represents and the field describing the state of the register (0--free, 1--occupied by a constant which must be indicated directly in the instruction, 2--occupied and there is a copy in the memory, 3--occupied and no copy in the memory, 4--register occupied and its contents cannot be altered). In turn, the triad contains a counter which indicates for how many triads it is an operand. Each time the triad is used as an operand in the generation of instructions its counter is reduced by a unit. When it becomes equal to 0 (i.e., the triad is no longer needed), the registers associated with it are marked as free. If an adder, for example, is needed for the next computation and none are free, the adder with the minimum value of the state field is occupied. Here attention is not paid to the counters in triads. Although theoretically this algorithm is not optimum, in practice it is totally satisfactory.

Let us present the following data. Sixteen FORTRAN programs with a total size of 5614 strings were translated. With this 28383 machine instructions were generated, 5235 of them being instructions for setting adders. In only 98 instances did a setting instruction cancel values whose presence in the register would be useful later on. In 27 of these for the purpose of freeing an adder it was necessary to copy it into the memory. In seven programs there was always a free adder and in 12 programs it was not necessary to copy adders into time variables. Data on the most "inconvenient" program: 579 strings, 2578 machine instructions, 530 adder settings, 17 of them (3.2 percent) canceled out needed values, and in all 17 cases it was necessary to copy the adder into the memory.

As an experiment the table of adders was reduced from eight to four elements. The register distribution algorithm remained totally workable: Of 5441 setting instructions 564 canceled out needed values. In 117 cases (2.2 percent of the total number of settings) it was necessary to copy the adder into the memory. The conclusion can be drawn that for a locally optimizing compiler the presence of eight adders renders unnecessary complicated algorithms for evading the triad graph and for the distribution of registers. In our opinion the optimization examples described in sec 6 can produce a great savings.

The distribution of descriptor registers is accomplished somewhat differently. The difference is that one descriptor can be used for access to several variables and arrays; therefore it is difficult to determine from counters in triads when a certain descriptor becomes unnecessary. In the generator described descriptor registers are marked as free only at the end of a line section. An analysis of 10 programs (3370 strings in FORTRAN, 18,459 machine instructions) demonstrated that of 3320 descriptor register loading instructions 170 (5.1 percent) spoiled information which later had to be put back into registers. (The instruction generator uses 11 descriptor registers.) On average the result is not bad, but the behavior of the most "inconvenient" program is considerably worse than the average: 103 out of 577 (17.9 percent) of instances of loading into descriptor registers destroyed needed values. This program contains accesses to subroutines with a great number of recurrent parameters. (In one line section are contained six accesses to a subroutine with 34 parameters, and 29 parameters of the last of the six accesses had been transferred previously.) With the available number of descriptor registers it is impossible to avoid completely the expulsion of required information, but knowledge of the moment when information in a register becomes unnecessary in this case would have reduced the number of "superfluous" loading events twofold (the total number of loading events would have been reduced by approximately 10 percent).

For the purpose of optimizing cycles, invariant and increment expressions (i.e., expressions which vary by a constant amount after each completion of a cycle) are computed before entering a cycle (in the cycle's prologue) and are placed in registers.

3. Distribution of Memory

The memory for local variables and arrays of the subroutine to be translated is assigned in a single segment insofar as this is possible. Then the following segment is used, etc.

Let us describe the algorithm in greater detail. Local variables are distributed in a segment in the following order:

1. Simple variables of the real type and other types requiring one or two words.

2. Simple whole-number variables.

3. Time variables of the CHARACTER type.

4. Simple logical variables and time variables of the BIT type.

5. Logic arrays, variables and arrays of the BIT type.

6. Variables and arrays of the CHARACTER type.

7. Whole-number arrays.

8. Arrays whose elements occupy not less than one word.

At the start of a program is placed the instruction for setting the base descriptor which indicates the point between variables and arrays; it has a word type of base. Simple variables and arrays are addressed with reference to this descriptor if it makes possible a 16-bit shift in the instruction. This distribution of variables is caused by the fact that in the AS-6 central processor the size of an instruction depends on the size of the shift in it. The size of the field for shifting in an instruction varies from 0 to 2 bytes depending on the size of the shift. This memory distribution arrangement increases the number of variables with slight shifts relative to the base descriptor.

As already mentioned, an individual segment is assigned for each common block. The subroutine's instructions and its constants are distributed in an individual segment. Writing into this segment is inhibited by hardware; therefore, the possibility of the erasure of instructions is eliminated, as well as the possibility of changing constants by means of assignment to formal parameters.

4. Accessing of Elements of Arrays

The AS-6 central processor instruction set makes it possible to program accessing of an element of an array by several methods. The instruction generator described uses only index registers for indexing and not half-word adders, indexing with which is performed more slowly. In addition, the instruction generator in the line section obviates the installation of new descriptor registers since operations with descriptors are performed rather slowly. However, in the cycle prologue descriptors are established, inasmuch as is possible, for all arrays which are worked with in the cycle.

5. Creation of Machine Instructions

AS-6 central processor instructions have various modifications. For example, in the majority of 2-operand instructions the first operand must be in a register and the second can be in the on-line memory, in a register or part of a register (e.g., in the left half of an adder for operations with half-words), and also can be a constant specified directly in the instruction (constants can be of two lengths--1 and 3 bytes). Instructions for jumps, the formation of descriptor registers, etc., have a special format. Accordingly, the writing of an instruction in autocode also has several formats. In view of this, the method of creating machine instructions used by us, making it possible to generate uniformly various sequences of instructions, can be of interest.

The GYeNKOM subroutine for the creation of machine instructions receives the following parameters from the triad processing subroutine: reference to the so-called macroinstruction, and several macroinstruction operands (their number depends on the macroinstruction). An operand can be the number of a register or a reference to a triad or to an element of a table of characters.

The macroinstruction consists of pictures of machine instructions. In a picture are described the operation (it is represented by a reference to a description of the machine instruction) and operands of the instruction in question. A certain operand can be the same for all uses of a given macroinstruction and then its value is presented in the picture. Otherwise the description of the operand in the instruction's picture indicates the number of the parameter of the GYeNKOM subroutine which must be put in place of this operand. The description of a machine instruction contains an operation code (the symbolic name of the instruction in autocode) and several more bits demonstrating how the features of the result are changed when the instruction in question is executed.

The GYeNKOM subroutine analyzes the macroinstruction and generates machine instructions, substituting in place of operands not specified in the macroinstruction values conveyed as parameters. Furthermore, if the parameter is a reference to a triad it is ascertained whether the value of the triad is found in a register and whether it is a constant which can be placed in the instruction. Various modifications of the instruction are generated depending on the fulfillment of these conditions.

6. Optimization in the Instruction Generator

The following actions are taken for the purpose of optimization in the instruction generator:

1. Jump instructions causing the transfer of control to the next instruction are eliminated.

2. Unneeded check instructions before conditional jumps are eliminated on account of the following of instructions which produce result control characters.

3. The commutativity of operations is employed, which makes it possible to avoid unnecessary instructions for loading into adder registers.

4. The available instruction for the reverse subtraction of numbers with a floating point is used together with the subtraction instruction.

5. Multiplication or division of a whole number by a power of two is replaced by an arithmetic shift.

6. Everywhere it is possible an operand which is a constant is placed directly in an instruction, which reduces the number of accesses to the memory.

7. When assigning a zero value, the zero write instruction is used, and when assigning a logical variable the value TRUE, an instruction for writing a 1 into the position.

8. Parameters for function-formulas are transferred through value and not through reference.

9. If assignments are not made for the parameter of a subroutine, the value itself is written into the place of the reference to the value of the parameter upon entering the subroutine.

4. Comparison with the FOREKS Compiler for the BESM-6

The main advantage of the AS-6 central processor is the large virtual memory, which makes it possible to describe in a natural manner problems whose programming on a BESM-6 causes serious difficulties. Still another plus is the much larger, as compared with the BESM-6, range of real numbers which can be represented and the hardware for working with double-precision numbers.

As far as working speed is concerned, it is not much better than in the BESM-6. Data which we have accumulated demonstrate that FORTRAN programs which do not employ operations with double precision are run on the AS-6 central processor an average of 1.2- to 1.5-fold faster than on the BESM-6. In problems which intensively employ whole-number arithmetic the gain in speed is higher (close to twofold). A threefold gain in speed was observed in logic tasks. Since the optimization procedures carried out are the same for both computers and the design principles of the instruction generators are close, the difference in running speed is caused basically by the different speeds of the BESM-6 and AS-6 central processor.

However, in certain cases a program is run more slowly on the AS-6 central processor than on the BESM-6. The reason for this is the small number of postscript registers. If in a cycle eight or more pages of the on-line memory are worked with a postscript will be retrieved constantly, which drastically reduces the speed.

Let us cite the following figures. One program is run on the BESM-6 in 3 min 30 s. On the AS-6 central processor it took 4 min 5 s to run. The fact was that in a short cycle nine arrays were worked with and they were all on different pages. When these 10 [as published] two-dimensional arrays were replaced by a single three-dimensional, as a result of which the elements which are worked with at close moments of time were side by side, the running time, in spite of the complication of access to elements of the arrays, was reduced to 2 min 25 s. The savings is quite considerable.

Thus, in writing a program it is necessary to pay attention to the fact that work at close moments of time be done with close addresses, i.e., that the working set be sufficiently small. This requirement relates to programming for any computer with a virtual memory. With frequent jumping from page to page, besides the retrieval of a postscript the transfer of data from the external storage into the on-line memory will often take place, which also slows down execution of the task.

As far as the size of the programs generated is concerned, the instruction portion of programs is approximately 20 percent shorter for the AS-6 central processor.

5. The AS-6 Central Processor as a FORTRAN Machine

The overwhelming majority of programs run on a computer are written in a high-level language, i.e., machine instructions themselves are generated not by a human being but by a compiler. Therefore, on the one hand in making a compiler every effort should be exerted to utilize the advantages of the machine and, on the other hand, in designing a computer it is necessary to pay attention to existing programming languages and methods of compilation, taking into account what the translator can do and what it cannot do, what is easy for it and what is very difficult. FORTRAN is one of the most common languages and it is important to us to evaluate the AS-6 central processor from the viewpoint of FORTRAN, for a programmer writing in this language is interested precisely in the characteristics of a virtual FORTRAN machine and not of a real physical machine.

On the whole the AS-6 central processor is quite successful as a FORTRAN machine. The virtual memory, the presence of a great number of directly addressable registers, the advanced instruction set, the unique state stack mechanism, the memory protection apparatus--all these represent advantages of no small importance.

It is necessary to mention that the AS-6 complex is designed for a broad range of applications. Therefore, if the AS-6 central processor is used only as a FORTRAN machine many capabilities of the processor will be unutilized. Of the 186 non-privileged instructions of the AS-6 central processor a FOREKS compiler can use only 80 (43 percent). For the sake of comparison, let us say that on the BESM-6 the FOREKS uses 35 instructions out of 46 (76 percent).

The program magazine (similar to the magazine in the BESM-6) available in addition to the state stack is utilized poorly. The powerful mechanism for the virtual performance of operations, when an operation is performed in the usual manner but the result is written nowhere but serves the purpose of generating control characters, is utilized only to a small extent. The FOREKS compiler can generate only two of these instructions: virtual setting and virtual subtraction. It seems to us that the other check instructions are also not necessary for a FORTRAN machine, and instead of virtual subtraction it is preferable to utilize a comparison instruction which does not result in overflow. Of course, in the AS-6 central processor there are instructions for comparing whole numbers, bytes and bits, but there is no instruction for comparing numbers with a floating point.

Meanwhile it seems to us that for the convenient implementation of a compiler from FORTRAN in the AS-6 central processor there are not enough setting and write instructions and instructions for dividing numbers with double precision. Also inconvenient is the fact that the addition, subtraction and multiplication of numbers with double precision can be performed only by having placed both operands in adder registers.

In [10] it is stated that a machine is good if a compiler can utilize its advantages without high costs. The AS-6 central processor does not totally meet this requirement. For example, it is difficult to select an optimum method for accessing elements of an array.

Of course, the AS-6 central processor offers many opportunities for optimization, the utilization of which is fairly simple (they are listed above), and this is an important advantage of the machine.

The results of a measurement of the frequency of the use of various instructions of the AS-6 central processor in the text of programs generated by a FOREKS compiler are given in table 1. The gathering and processing of statistical data were performed by means of the POPLAN system [11]. Several FORTRAN programs numbering a total of 2146 instructions were translated. With this 12,754 machine instructions were generated. Fifty-four instructions of the AS-6 central processor were used.

Table 1.

Name of instruction	Percentage of use
Formation of descriptor	14.5
Writein of descriptor	11.4
Setting half-word adder	8.6
Setting adder	7.4
Writein of adder	6.6
Writein of half-word adder	5.5
Setting saving mask	4.9
Jump with return	4.9
Setting index register	4.5
Multiplication with floating point	4.0
Setting descriptor with connective	4.0
Addition of half-word whole numbers	3.5
Addition with floating point	2.8
Index register writein	1.8
Setting descriptor	1.8
End of cycle	1.6
Subtraction with floating point	1.5
Multiplication of half-word whole numbers	1.3
Write zero into word	1.1
Other instructions	8.4

The most used were the instructions for forming and writing in descriptor registers. They are used mainly in transferring parameters. The formation instruction is rather slow, which has a negative effect on the running speed of FORTRAN programs.

Let us point out also that virtual operations made up only 0.3 percent. Conversion of a number from a whole number into a floating was encountered 41 times and from a floating into a whole 5 times. The reverse subtraction instruction made up 20 percent of all cases of subtraction with a floating point. It was possible to use the shift instruction in 32 percent of instances of the multiplication and division of whole numbers. The writein of a zero made up 15 percent of all word writein operations.

Bibliography

1. Shtarkman, V.S. "Svravnitel'nyy analiz translyator FOREX" [Comparative Analysis of FOREX Compiler], Preprint No 129, USSR Academy of Sciences Institute of Applied Mathematics, 1978.

2. Shtarkman, Vik. S. "Lokal'naya optimizatsiya ob"yektnoy programmy v translyatore Foreks" [Local Optimization of an Object Program in the FOREKS Compiler], Preprint No 149, USSR Academy of Sciences Institute of Applied Mathematics (IPM AN SSSR), 1979.

3. Mikhelev, V.M. and Vershubskiy, V.Yu. "ASTRA. Novyye vozmozhnosti yazyka" [ASTRA: New Possibilities of the Language], Preprint No 61, IPM AN SSSR, 1976.

4. Chaykovskiy, M.G. et al. "Avtokod dlya tsentral'nogo protsessora sistemy AS-6" [Autcode for the AS-6 Central Processor System], ITEF-153, Moscow, 1976.

5. "Draft Proposed ANS Fortran (X3J3/76)," SIGPLAN NOTICES, Vol 11, No 3, Mar 1976.

6. "Programmirovaniye na yazyke Assemblera YeS EVM" [Programming in the YeS Computer Assembly Language], Moscow, Statistika, 1976.

7. Tsikritzis, D. and Bernstayn, F. "Operatsionnyye sistemy" [Operating Systems], Moscow, Mir, 1977.

8. Gris, D. "Konstruirovaniye kompilyatorov dlya tsifrovykh vychislitel'nykh mashin" [Designing Compilers for Digital Computers], Moscow, Mir, 1975.

9. Aho, A.V., Johnson, S.C. and Ullman, J.D. "Code Generation for Expressions with Common Subexpressions," JACM, Vol 24, No 1, Jan 1977.

10. Wirth, N. "The Design of a PASCAL Compiler," SOFTWARE - PRACTICE AND EXPERIENCE, Vol 1, No 4, Oct 1971.

11. Bayakovskiy, Yu.M., V'yukova, N.I., Galatenko, V.A. and Khodulev, A.B. "Programmirovaniye na yazyke POPLAN" [Programming in POPLAN], IPM AN SSSR, 1976.

8831
CSO: 1863/45

UDC 681.3

SYSTEM OF DIALOGUE PREPARATION OF TASKS FOR UNIFIED SERIES COMPUTERS

Moscow PROGRAMMIROVANIYE in Russian No 5, Sep-Oct 81 (manuscript received 2 Sep 80) pp 68-73

[Article by V.V. Khanykov, A.V. Rybakov and N.V. Anan'ina]

[Text] In this paper a description is given of an instrument system for working with a design library for Unified Series (YeS) computers in the operating system.

The development, debugging and utilization of software in a YeS operating system assume the heavy use of an assignment control language (YaUZ). The complexity of the software, the English language mnemonics and the "machine" orientation of the YaUZ cause a great number of errors in assignments. The apparatus of catalogued procedures only partly facilitates the user's situation.

For the development of YeS computers the problem of facilitating communication between the user and the YeS operating system assignment control system on the national level has become pressing.

In this paper a solution based on the creation of a psychologically natural input language is discussed and a description is given of the implementation of a preprocessor for the dialogue preparation of assignments for YeS computers.

The rejection of a dialogue assignment remote input (DUBZ) system was caused by the fact that the instruction language used in it requires a great number of input characters in describing an assignment and is oriented toward the English language. In addition, when working with a DUBZ system the possibility of parallel-series input is lacking and the effectiveness of the work itself is determined to a great extent by the configuration of the system and the number of users working simultaneously.

The features of the development of the input YaUZ by means of a preprocessor are illustrated in the example of the creation of a function-oriented language for working with a design library [1]. The procedure for specifying function statements is as follows:

An analysis is made of user's actions in the development and debugging of routines by means of a design library.

In the user's work typical actions are singled out which are usually independent of the computer and are determined by the specifics of the area of analysis.

Each typical action or combination of them is assigned some function-oriented statement whereby the variable parts of these statements correspond to real objects on which the user plans actions in his assignment.

The following can be placed under the heading of typical actions of a user when working with a design library: designation of a specific library from a great number of permissible libraries in the system, selection of a specific section of a library, and the set of functional operations relating to manipulating a selected object. The minimum set of function-oriented statements for working with a design library must include the following statements: start of assignment, notation in design library, translation from library, editing of contents of library, start execution, circulation of contents of library for different machine media, etc. The variable parts in these statements specify the names of assignments, libraries and objects in the system.

For convenience of the user's working with the system a set of service functions must be provided for the creation of the design library and for maintaining it in the working state. In addition, the service functions provide the user with all the necessary information regarding the state of the design library and the system as a whole.

The procedure for analyzing YeS computer YaUZ statements for the user's work with the design library and for the formation of function-oriented statements is discussed in [2].

A set of Unified Series operating system assignment control language (YaUZ OS YeS) standard statements performing similar actions is specified for each function-oriented statement. The software converting function-oriented statements into standard assignment control language statements performs the role of a preprocessor for the YeS operating system.

The purpose and functions of the steps of preparing and performing an assignment in the process of the creation of software are different. At the first step the user must formulate the sequence of actions in the assignment and describe it in the assignment control language. At the second step these actions are implemented in the computing environment of a YeS computer.

The authors sugggest that minicomputers be used in the first step. For this purpose a dialogue assignment preparation (DPZ) system has been developed for YeS computers. The main objective of the system is for the user to be able to use function-oriented statements in the dialogue mode for working with the design library, as well as to offer him timely consultation in working with the assignment control language. Here it is necessary to make possible the recognition of function-oriented statements regardless of the national language.

The structure of the software of the dialogue assignment preparation system was chosen on the basis of this objective (fig 1). The system can be in one of two modes--"Assignment Preparation" or "Consultation"--whereby the first is the basic mode.

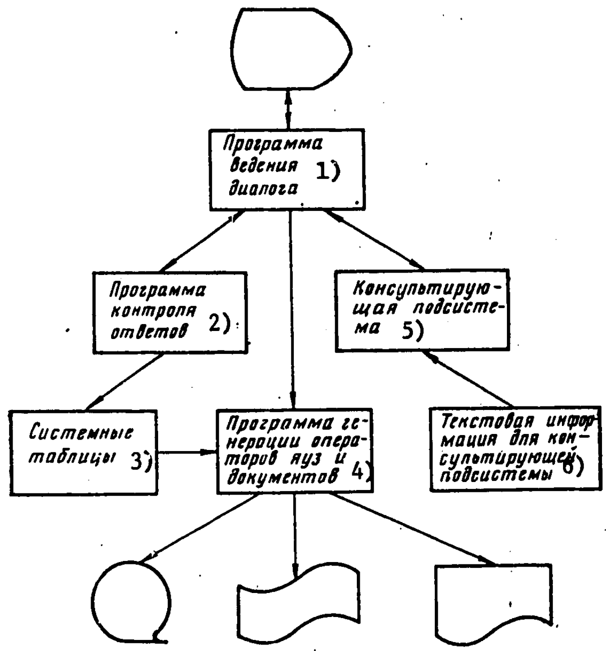


Figure 1.

Key:
1. Dialogue maintenance routine
2. Response check routine
3. System tables
4. Assignment control language statement and document generation routine
5. Consultation subsystem
6. Text information for consultation subsystem

When it is necessary to obtain information the user can establish the "Consultation" mode. For this he must begin the input of a response to any dialogue assignment preparation question with a special character.

In the system is provided the issuance of individual information on the purpose and syntax of variable information contained in function-oriented statements, on the algorithm for user's actions in the development and debugging of routines, and also on working with the dialogue assignment preparation system itself.

The changeover to the "Assignment Preparation" mode occurs automatically after the end of a response to a user's question. The existence of the two modes makes it possible for an untrained user not only to prepare packages of assignments independently, but also to learn the YeS computer operating system assignment control language.

Not the syntactical analysis of function-oriented statements but selection according to the type of function (selection from a list) is used in the dialogue assignment preparation system. The table of functional operations on the screen of the display is shown in fig 2. This method of implementing the recognition of function-oriented statements makes possible maximum independence from the national language and reduces the number of errors in writing an assignment. In addition, the problem of informing the developer regarding statements used in the dialogue assignment preparation system is solved automatically in this case.

PRINTOUT TEXT	GET INFORMATION	CLEAR AREA ON DISK
PRINT HEADING OF LIBRARY SL	PRINT HEADING OF LIBRARY RL	PRINT HEADING OF LIBRARY CL
COMPRESSION OF LIBRARY SL	COMPRESSION OF LIBRARY RL	COMPRESSION OF LIBRARY CL
REMOVAL FROM LIBRARY SL	REMOVAL FROM LIBRARY RL	REMOVAL FROM LIBRARY CL
TRANSLATE INTO PL/1	TRANSLATE INTO FORTRAN	TRANSLATE INTO ASSEMBLY LANGUAGE
CHANGES	PRINTOUT VTOC	EXECUTE
CATALOGUING IN LIBRARY SL	OPTIMIZATION OF ASSIGNMENT	CATALOGUING IN LIBRARY CL
END OF FORMATION OF ASSIGNMENT	PRINT ENTIRE LIBRARY	INDEPENDENT OPERATION

Figure 2. [Designations SL, RL, CL and VTOC are in the Roman alphabet; the rest is in Russian.]

After the user selects a function the system initiates a dialogue, specifying refining questions in the natural language relating to objects and the attributes of specific statements. The dialogue between the user and the dialogue assignment preparation system is constructed on the basis of rules for the use of function-oriented statements.

The user's responses in the "Assignment Preparation" mode pass tests based on the requirements of the syntax of the YeS operating system assignment control language. In the case of an incorrect response on the part of the user the dialogue assignment preparation system diagnoses the error and suggests that he repeat the input of a response. Only correct responses are entered in the appropriate table of the dialogue assignment preparation system.

A table with the list of functions lights up on the display's screen each time after the end of the input of the required information relating to the preceding function. This makes it possible to construct packages of assignments not according to a strict algorithm but in keeping with the user's requirements.

The purpose of the "Optimization," "Get Information" and "Independent Operation" functions of the dialogue assignment preparation system must be emphasized especially. When it is necessary to optimize an assignment with respect to the time for execution on a YeS computer and/or with respect to the distribution of the magnetic disk storage, the "Optimization" function must be indicated as the last function in creation of the package. The "Get Information" function accomplishes the output to the display's screen and alphanumeric printer of the basic key concepts used in working with the dialogue assignment preparation system. The "Independent Operation" function makes it possible for the user to form directly on the display's screen the YeS operating system assignment control language statements needed by him. Thereby the conditions are created for describing unique assignment steps, which makes it possible to realize the multifunctional capabilities of a YeS computer operating system.

Upon termination of the formation of an assignment the dialogue assignment preparation system, based on the information stored in system tables, constructs YeS operating system assignment control language statements and the documentation for the package created. Instructions for the computer operator and a verbal description of assignment steps are included in the documentation.

An example of a dialogue between a user and the dialogue assignment preparation system is shown in fig 3. An assignment for a YeS operating system (fig 4) was formed as the result of the system's operation. The parallel-series input of information is provided in the dialogue assignment preparation system for the experienced user. This makes it possible for the user to input, in response to the first question, all the necessary information relating to the function selected.

DPZ [dialogue assignment preparation system]:	DPZ SYSTEM WORKING. ENTER NAME OF ASSIGNMENT.
USER:	RABOTA [in Roman alphabet]
DPZ:	ENTER CLASS OF ASSIGNMENT.
USER:	?
DPZ:	CLASS OF ASSIGNMENT DETERMINES PRIORITY OF RUNNING OF ASSIGNMENT. SPECIFIED BY ROMAN CHARACTERS FROM A TO O.
USER:	A
DPZ:	ENTER NAME OF LIBRARY.
USER:	PRO_BA
DPZ:	ERROR IN ENTRY OF LIBRARY NAME. REPEAT ENTRY.
USER:	PROBA [in Roman alphabet]

* * * *

Then a table with a list of functions is output onto the display's screen. The user indicates by means of the cursor the function "Cataloguing in SL."

* * * *

[Continued on following page]

DPZ: ENTER NAME OF TEXT.
USER: TEXT [in English]
DPZ: ENTER DATA.
USER: TEXT: PROCEDURE OPTIONS(MAIN); [in English]

* * *

USER: /*

* * * *

A table with a list of functions is output onto the display's screen. The user indicates by means of the cursor the function "Translate into PL/1."

* * * *

DPZ: ENTER NAME OF MODULE.
USER: TEXT [in English]

* * * *

A table with a list of functions is output onto the display's screen. The user indicates by means of the cursor the function "End of Formation of Assignment."

Figure 3.

```
//RABOTA JOB MSGLEVEL=(1,1),CLASS=A  [in English]
//*
//* CATALOGUING IN LIBRARY [in Russian] SL
//*
//RAB0001 EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DSNAME=PROBASL,
//   UNIT=SYSDA,DISP=MOD,VOLUME=SER=OSVALD
//SYSIN DD DATA
./ ADD NAME=TEXT,LIST=ALL,LEVEL=00,SOURCE=0
./ NUMBER NEW1=0,INCR=1
 TEXT:  PROC OPTIONS(MAIN);
.
.
.
/*
//*
//*  TRANSLATE INTO PL/1  [in Russian]
//*
//*  STEP1 - TRANSLATION [in Russian]
//*
//RAB0002 EXEC PGM=IEMAA,PARM=NODECK,LOAD,SIZE=52K',
```

[Continued on following page]

```
//   REGION=52K
//SYSPRINT DD SYSOUT=A
//SYSUT3 DD DSNAME=&&SYSUT3,UNIT=SYSDA,
//   SPACE=(80,(250,250)),
//   DCB=BLKSIZE=80
//SYSLIN DD DSNAME=&&LOADSET,DISP=(MOD,PASS),UNIT=SYSSQ,
//   SPACE=(80,(250,100))
//SYSUT1 DD DSNAME=&&SYSUT1,UNIT=SYSDA,
//   SPACE=(1024,(60,60),,CONTIG),
//   SEP=(SYSUT3,SYSLIN),DCB=BLKSIZE=1024
//SYSLIN DD DSNAME=PROBASL(TEXT),
//   UNIT=SYSDA,DISP=OLD,VOLUME=SER=OSVALD
//*
//*  STEP2 - CATALOGUING IN LIBRARY RL  [in Russian]
//*
//RAB0003 EXEC PGM=IEWL,PARM='NCAL',COND=(8,LT,RAB0002)
//SYSPRINT DD SYSOUT=A
//SYSLMOD DD DSNAME=PROBASL(TEXT),
//   UNIT=SYSDA,DISP=OLD,VOLUME=SER=OSVALD
//SYSUT1 DD DSNAME=&&SYSUT1,UNIT=SYSDA,SPACE=(1024,(100,25)),
//   SEP=SYSLMOD, DCB=BLKSIZE=1024
//SYSLIN DD DSNAME=&&LOADSET,DISP=(OLD,DELETE),UNIT=SYSSQ
/*
//
```

Figure 4.

The dialogue assignment preparation system is implemented as a set of individual subroutines written in FOKAL. Therefore the addition of a new or the exclusion of an old function in the structure of the dialogue assignment preparation system does not present special problems and can be performed by the user.

The following set of hardware is needed for the normal operation of the dialogue assignment preparation (DPZ) system: a minicomputer of the "Elektronika" type (or a CM-3 or CM-4) with an expanded on-line memory unit; a VIDEOTON-340 display; an alphanumeric printer of the DZM-180 type; and an IZOT 1370 magnetic disk storage.

In conclusion let us formulate the distinctive features of the system suggested for the dialogue preparation of assignments for YeS computers.

1. The use of the national language in the dialogue is conducive to freeing the user from remembering complicated YeS computer assignment control language mnemonics and to reducing the time for the preparation of an assignment.

2. The existence of two modes makes it possible for the user to obtain a consultation at any moment of time in working with the system.

3. Checking of the correctness of user's responses eliminates syntactical errors in the preparation of an assignment, which is conducive to the efficient utilization of machine time for YeS computers.

4. The number of characters to be input by a user in working with the DPZ system is reduced drastically.

5. Adaptation of the DPZ system to the level of the user's professional training is made possible by the employment of a serial or parallel-serial data input mode, as well as by means of the "Independent Operation" function.

6. The functional nature of subroutines makes it possible to expand and develop the DPZ system without a cardinal change in its structure.

7. The tabular organization of data in the system makes the following possible: the entry of changes in information entered without waiting for the end of the formation of a package; the efficient construction of the sequence of assignment steps in a package; and the creation of uniform documentation conforming precisely to the assignment package, simultaneously with construction of the package itself.

Bibliography

1. Baker, F.T. "Chief Programmer Team Management of Production Programming," IBM SYST. J., Vol 11, 1972, p 56.

2. Anan'ina, N.V., Rybakov, A.V. and Khanykov, V.V. "Rationalization of a User's Work with a YeS Disk Operating System in the Development of Programs," ELEKTRONNAYA TEKHNIKA, Ser. 9, No 1, 1980, p 35.

8831
CSO: 1863/45

UDC 681.3.06

ORGANIZATION IN DISPAK OPERATING SYSTEM OF DETERMINATE OUTPUT OF INFORMATION OVER ENTIRE FIELD OF OUTPUT UNITS OF MULTIMACHINE COMPUTING COMPLEX

Moscow PROGRAMMIROVANIYE in Russian No 5, Sep-Oct 81 (manuscript received 28 Apr 80) pp 88-91

[Article by V.P. Petlinskiy and V.F. Tyurin]

[Text] Questions are discussed, relating to the software simulation by means of an operating system of the total field of output units in a multimachine computing complex. A method is suggested for determinate output based on dividing the single common output waiting line into a number of separate waiting lines. The implementation of this method in the DISPAK operating system is described.

In the development of operating systems (OS's) for a multimachine computing complex (MVK) traditional strategies and principles are often used for organizing the operation of both the entire system and of individual components of it in a single machine. This results in the fact that many additional possibilities of the MVK are unintentionally constricted or are not utilized at all, both from the viewpoint of conveniences for the user and of the more intelligent organization of running a solution to problems, and from the viewpoint of the effectiveness of the utilization of equipment.

For example, the traditional strategy for outputing information in the single-machine variant of the DISPAK operating system is buffering in the calculation process, formation of a request for output and placing it in a single common waiting line for the end of the task and attending to requests, uniformly formulated and indistinguishable from the viewpoint of the operating system, from the common waiting line after the end of the task. Let us note that a request formulated for output, after being placed in a common waiting line, does not carry any additional information regarding its belonging to a specific task of the user. This sort of organization of output has become widespread in many operating systems, since it makes it possible easily to implement a multiprogram calculation mode and is classified as system output [1].

With system output different kinds of information intended for output to various types of peripheral output units are distributed in the DISPAK operating system into their own output waiting line [2], and in the OS/MFT, OS/MVT and YeS [Unified Series] operating system [3, 4] into their own output class, similar to the output waiting line in the DISPAK operating system. One or more output units oriented toward a specific type of information to be output are connected to each output waiting line. Thus, in the output buffer it is possible to organize as many waiting lines as the types of output units specified in generation of the operating system in a given equipment configuration.

In the DISPAK operating system it is possible to indicate four types of output units in generation of the operating system for a single computer: the ATsPU-128/2M [alphanumeric printer] type for rapid printout (up to two); the PI-80 type for punched cards; the PL-80 type for punched tape; and the CALCOMP type for a graph plotter [5]. In the process of this study in the output buffer it was possible to organize up to four individual output waiting lines for the number of types of output units.

Let us discuss the procedure for attending to each of these waiting lines. Requests from the waiting line are selected for service according to the rule "first come, first served" (FIFO [first in, first out] [6]). If not just one unit works with the waiting line in question (output waiting line for rapid printout), then units are selected for operation in cyclic order, by turns [2]. It is obvious that in the latter case according to the arrangement described above for the organization of system output, the possibility of tying in the output from a specific task to a specific unit--determinate output--is eliminated. The same output strategy was used also in DISPAK operating system versions at work in BESM-6 multimachine computing complexes.

Meanwhile the existence of a common output buffer for the MVK makes it possible to consider the combination of all output units of the complex as a common output field (a maximum of eight ATsPU-128/3M's, four PI-80's, four PL-80's and four CALCOMP-type graph plotters), but not with the hardware switching of units [7] for operating with a required computer of the complex, but by means of software switching or the software simulation of this switching. With this more flexible strategies for utilizing each unit in the total field are possible. The need for these strategies is occasioned by the large flow of information output from all computers of the complex requiring determinate output in the total field, as well as by the untraditional methods of processing output information in the dialogue terminal systems undergoing intense development in recent times.

These strategies must make the following possible:

The establishment of priority service according to the "last come, first served" rule (LIFO [last in, first out] [6]) for specific categories of users.

The putting together of a continuous printed output listing (a roll) for the tasks of one or more subdivisions of users for a specific time period--the administration of output.

Distribution of the output of users of the teleprocessing system over remote user stations furnished with rapid printing units.

Storage of the output of specific tasks of users on a disk (tape) in the form of an output file, but without actual printout for the purpose of subsequent processing in the dialogue mode at a terminal.

With the output of especially valuable information (e.g., in the machine duplication of documentation on a unit with high printing quality), the sending of it to a unit set aside for this purpose.

Checking the operation of the required output unit by means of test tasks while solving other problems in keeping with a set schedule without disrupting the usual solving mode.

For the purpose of enabling the capabilities indicated above, the output strategy in the DISPAK operating system described above was changed. One of the main principles on which the new strategy is based is the separation of the single common output waiting line for each type of information into a common line and a number of separate output waiting lines (a total of 32 waiting lines for each type of information). The second important principle is the introduction of software-simulated switching of any of the output units of the total output field of the MVK for working with only one of the output waiting lines.

With this output strategy each separate waiting line receives requests for the output of specific tasks of users, and each waiting line is served by one (or more) preindicated unit. Additional tables of two types placed in the common disk storage have been introduced for the purpose of implementing this kind of service.

The first type of table is represented by tables for the selection of requests (from a combination of key characters of the user's code) and for the distribution of requests selected to the required output waiting line. In the same tables information is stored on the order for placing a request in the waiting line for service (FIFO or LIFO). The second type of table--the unit switching table--contains information on which waiting line is served by a specific unit of the MVK's total output field. Tables of both types contain check information which is used for checking their contents from the entry of a random code both with computer hardware errors and with possible software errors.

The employment of the method described above for organizing determinate system output has a number of important advantages over the alternate method implementing these possibilities through the system of working with a single common output waiting line. It is obvious that in this case each request sent for buffering to the common waiting line must carry an identifier of the unit out of the entire combination of units assigned to this waiting line which will receive the actual output. Later, when the common output waiting line is dumped, this identifier is used for referencing to a specific unit.

It is possible to single out three key advantages of the output strategy used. First, the tabular method of distributing information over separate waiting lines makes it possible to achieve great flexibility in possible distribution variants at a lower cost, since there is no longer a need to revise operating system routines for the purpose of adjusting for the variant selected. For the required distribution it is sufficient to fill in the tables in the external storage anew, which is easily achieved by using service utilities.

Second, in the distribution of addressed requests from a common waiting line the need arises of retrieving, sorting and accessing requests from the waiting line according to individual identifiers of units. The addition of these algorithms can increase to a great extent the amount of routines in the operating system. Meanwhile the specifics of the running of operating system routines (strict requirements with regard to the capacity of the memory used) reduce and in our case totally eliminate the possibility of using algorithms of this sort. Algorithms with preliminary tabular distribution with subsequent simple serial accessing are implemented considerably more compactly.

Third, in our case overhead costs in the operating system for the startup of schedulers of the operation of external units are reduced, since it is known beforehand whether there are requests for operation in the waiting line to which the external units of a given computer of the multimachine complex have been assigned. In the variant which is the alternate of this, startup of the external unit operation scheduler must be carried out whenever the common line is not empty, since its functions include the sorting of requests and referencing to the required unit.

It can also be stated that for the output strategy employed the protection of information from unsanctioned access and the LIFO rule for privileged requests for output are implemented easily.

All 32 output waiting lines have a different status. The waiting line with the number 1 is the common waiting line. It receives all requests for output which are not distributed by means of selection tables to any separate waiting line. Waiting lines with numbers beginning with 2 and ending with 20 are transient or on-line planning waiting lines. The sequence for the placement of requests in these waiting lines and the serving of requests from these lines can be altered on line on the basis of up-to-the-minute requirements. The waiting line with the number 13 is used for the distribution and accessing from it of requests for test output, for testing the proper working order of the required unit of the MVK's total output field. The waiting line with the number 14 is used for buffering requests for output which are later to be teleprocessed. The copying of information from this waiting line onto a teleprocessing disk (tape) is provided for in the system. The placement of requests into this waiting line is possible both by the usual method--by means of request selection-distribution tables--and through an extracode for storage of the output file in a teleprocessing disk (tape). The storage of these files in a general archive of output teleprocessing files has been implemented at the present time. The organization of personal archives of user's output files is to be provided for later. This method can be used as an alternative method of the sheet-by-sheet interrogation of output information described in [8]. Distribution into this waiting line for administrative requirements also has been provided for tasks which end accidentally. Waiting lines with the numbers 15 and 16 are reserved for serving remote teleprocessing stations furnished with rapid printout from output units of the total MVK field. Waiting lines numbering 17 through 32 are administrative. The placement of requests for output and the output of information from these waiting lines are accomplished automatically. Furthermore, in units set aside for serving administrative waiting lines the mode of outputing printouts for user subdivisions has been implemented. All the information of a schedule for a fairly long time period (up to 2 to 3 24-h periods) is assigned in advance and is stored in system tables and scales. This information determines groups of users to be served in this mode, the units on which the serving of administrative waiting lines is implemented, and the following order of periods for serving user subdivisions.

The protection of information distributed into any separate waiting line (through an access key) and protection from its redistribution into another waiting line are provided for in the system. The same protection is provided also for units attached for serving a specific waiting line.

Of course, the formation and correction of tables and scales for all five types of waiting lines is a labor-intensive process which is difficult to perform by hand without errors. The SYeRB (Servis Razgruzki Bufera [buffer dumping service]) dialogue system was developed for facilitating the entire set of operations relating to enabling determinate output capabilities in the total field of MVK output units. This system includes a number of utilities which make it possible to obtain information on all waiting lines in a form convenient for perception on the screen of a display, to control on-line waiting lines, to organize archives of output files, to form, edit and enter into system tables a schedule for administrative waiting lines, and to set and cancel access keys for waiting lines.

An instruction language is used for communication with users of the SYeRB system. Protection from unsanctioned access to the system through a key is provided for in the SYeRB system.

Proposed as a further development of the system described is the creation of a combination of utilities for working with archives of user output files for the purpose of their dialogue or package editing and output in processed form to the required unit.

In conclusion let us mention that all the above-described capabilities of the DISPAK operating system and SYeRB system have not been realized in operating systems known to us both for the BESM-6 computer and computers of the YeS type [4, 9]. On the other hand, the organization of system output in keeping with the system described in this article can be used with success both for BESM-6 MVK's and YeS MVK's, as well as for MVK's of greater capacity.

Bibliography

1. Madnik, S. and Donovan, Dzh. "Operatsionnyee sistemy" [Operating Systems], Moscow, Mir, 1978.

2. Zel'dinova, S.A., Koshkina, L.V., Ozorin, Yu.V., Tyurin, V.F. and Shulepov, N.I. "Structure and Functioning of the DISPAK Operating System," TRUDY CHPI, No 136, Chelyabinsk, 1973.

3. Kattsan, G. "Operatsionnyye sistemy (pragmaticheskiy podkhod)" [Operating Systems, a Pragmatic Approach], Moscow, Mir, 1976.

4. Peledov. G.V. and Raykov, L.D. "Vvedeniye v OS YeS EVM" [Introduction to the Unified Series Computer Operating System], Statistika, 1977.

5. Tyurin, V.F., ed. "Operatsionnaya sistema DISPAK dlya BESM-6 (sistemnomu programmistu i operatoru)" [DISPAK Operating System for the BESM-6 (for System Programmers and Operators)], Moscow, IMP AN SSSR [USSR Academy of Sciences Institute of Applied Mathematics], 1973.

6. Tsikritzis, D. and Bernstayn, F. "Operatsionnyye sistemy" [Operating Systems], Moscow, Mir, 1978.

7. Enslou, F.G. "Mul'tiprotsessornyye sistemy i parallel'nyye vychisleniya" [Multiprocessor Systems and Parallel Computations], Moscow, Mir, 1976.

8. Zel'dinova, S.A., Paremskiy, M.V. and Tyurin, V.F. "Nekotoryye bazovyye vozmozhnosti OS DISPAK" [Some Basic Capabilities of the DISPAK Operating System], Moscow, IPM AN SSSR, 1976.

9. Lomidze, O.N. and Silin, I.N. "Automated System for Buffering Results of the Calculation of Problems Utilizing Magnetic Disks in the 'DUBNA' Operating System for the BESM-6 Computer," PROGRAMMIROVANIYE, No 3, 1978, p 56.

8831
CSO: 1863/45

PUBLICATIONS

UDC 629.7.017.2

CONTROL ALGORITHMS FOR SPACECRAFT

Moscow UPRAVLENIYE DVIZHUSHCHIMISYA OB"YEKTAMI NA OSNOVE ALGORITMA S MODEL'YU in Russian 1981 (signed to press 17 Apr 81) pp 2-8, 227-232

[Annotation, introduction, bibliography and table of contents from book "Control of Moving Objects Based on an Algorithm with a Model", by Igor' Mikhaylovich Sidorov, Lyudmila Yevgen'yevna Goncharova and Valeriy Georgiyevich Lebedev, Izdatel'stvo "Mashinostroyeniye", 1063 copies, 232 pages]

[Excerpts] Annotation

This book reviews a new class of algorithms with models that are realizable in onboard computers included in the control contours of moving objects such as flying craft, the rolling stock of high-speed surface transportation systems, and other complex mechanical systems. It considers algorithms with models that have a high level of adaptability to external disturbances and to a situation of limited information. The potential of the algorithms described is demonstrated through the example of a number of problems of orienting and stabilizing controlled objects. The book is intended for engineers specializing in dynamics and control of flying craft.

Introduction

The use of onboard digital computers in the control systems of rockets and spacecraft raises a number of new questions, among which are the designing and building of the computers themselves and their associated input and output data convertors. Another set of problems comprises the synthesis of control algorithms that can be realized on such computers and analysis of the quality of the processes of regulation in the closed system of the object and the regulator.

This book will consider only the questions of synthesizing algorithms to stabilize objects with complex dynamic schemes.

Including an onboard computer in the control circuit opens up new possibilities in building stabilization systems that would be difficult to realize on the basis of analog equipment. The use of onboard digital machines makes it possible to replace a number of stages required to snythesize a stabilization system with analog elements by forming a computer procedure for the stabilization algorithm. In this sense formulation of the algorithm, which ultimately is a program for realization on the onboard digital computer, should be viewed as a stage in designing the stabilization system.

But despite the fact that the traditional process of design is replaced by the selection of an algorithm where an onboard digital computer (ODC) is used, this does not mean that the problem of synthesizing this algorithm can be solved by using a formalized computing procedure. On the contrary, construction of a stabilization algorithm that is realizable on an ODC opens up the possibility of making fuller use of the characteristics of the dynamic scheme of the object being regulated.

In this book the stabilization algorithms are constructed for complex objects which include the equations of connected-in oscillators in their dynamic schemes. For objects whose equations of movement correspond to the movement of an object as a "solid state," the use of an ODC in the structure of the stabilization system can also produce a certain effect, related to the feature of using computing technology. In this case questions such as the variability of the parameters of the stabilization system, drift, and the like, which are very important for analog technology, are practically excluded from consideration.

In an ODC the restructuring of the parameters of the algorithm or a switch from one algorithm to another can be accomplished quite simply. These advantages are common to the construction of any type of stabilization system for an object when ODC's are used. As for the problem of synthesizing the stabilization algorithms, the principles of construction and methodology for selecting the parameters of the stabilization algorithm have been worked out for objects with simple dynamic schemes, and therefore it makes sense to use analog algorithms that have been transferred to ODC's.

In this case well-known methods of the theory of discrete systems can be used for analytic study of the closed system, the object and the regulator.

A greater impact should be expected from the use of ODC's to stabilize large objects with complex dynamic schemes. In this case the use of ODC's is justified both from an economic standpoint and considering the fact that the introduction of new technology should lead to expanded possibilities of solving new problems.

Stabilization algorithms realized on ODC's make it possible to more fully meet the requirements made by the object for a stabilization system than where analog technology is used.

The requirements which the object makes for its stabilization system should be worked out on the basis of a comprehensive study of disturbing forces and moments acting on the object, analysis of interference in the sensors that determine the coordinates of the object, and a detailed description of the dynamic scheme of the object. Compiling the dynamic scheme of the object includes solving problems of hydrodynamics that describe the oscillations of the liquid in the fuel tanks, problems of elasticity theory which describes flexural oscillations of the body of the object, and a description of the dynamics of the actuating organs of the stabilization system with due regard for their basic nonlinearities. It may be necessary to consider additional questions, for example to study the longitudinal oscillations of the object and the interrelationship between oscillations of the object in the stabilization planes and longitudinal oscillations.

The requirements made of the stabilization system are formulated on the basis of experience studying objects of different classes. Numerous monographs have summarized the results of these studies [1, 16].

Most of the published works devoted to the theory of optimal control concentrate their attention on construction of an optimal law of stabilization which insures minimum quadratic deviations of the coordinates of the object given the statistical characteristics of interference and the external forces and moments acting on the object.

Experience with the study of the dynamic schemes of various objects shows, however, that when a stabilization system is being built , in addition to influences on the selection of parameters of the system of external actions and interference the chief difficulties in selecting the structure and parameters, which are determining for the makeup of the control system, depend on internal factors. The characteristics of the object's dynamic scheme, which includes a series of connected oscillators, are determining.

The requirements made of a stabilization system vary in nature and often conflict with one another. Therefore, the process of designing the algorithm involves successively meeting a series of requirements, and cannot be reduced to the problem of constructing an algorithm that optimizes a chosen all-embracing criterion in some particular way.

The structure of a stabilization system using an ODC is set forth most fully in [2], where it is shown that the ODC permits more effective synthesis of a stabilization system for objects with complex dynamic schemes than does the use of continuous analog units. This work is able to trace the analogy in the calculations of discrete and continuous systems; this makes it possible to use the experience gained in the use of continuous systems for analysis and synthesis of discrete control systems. The amplitude-phase frequency characteristics of the discrete stabilization algorithm obtained in this way are close to the corresponding characteristics of an algorithm realized on continuous units. It should be underlined that the method of synthesizing a stabilization system for objects with complex dynamic schemes that is presented in work [2] is based on successive satisfaction of the requirements made of the stabilization system.

In the initial stage of design the basic parameters of the stabilization system are selected: the amplification factor and differentiation time constant for the object whose dynamic system has been described in simplified terms by solid state equations without considering the attached oscillators. The parameters of the stabilization system are selected so as to insure the necessary quality of regulation processes when the object is acted on by disturbing forces and moments. The determining factor in selecting the values of the parameters of the stabilization system is not the random components of the forces and moments, but rather the standard set of disturbing influences. This set describes the extreme values of the forces and moments, the gradients of disturbing influences, fixed initial conditions, the constant values of forces and moments resulting from maximum possible misalignment of the thrust of the engines, nonsymmetrical distributions of weight within the objects, and the like.

In the next stage the requirements for the amplitude-phase fequency characteristic of the stabilization system are fulfilled on frequencies that correspond to the oscillations of the attached oscillators.

The specific features of a stabilization system realized on continuous units and the fairly dense spectrum of frequencies and spread of the parameters of the dynamic system of the object away from nomimal values, which is an especially important consideration when designing a stabilization system, result in a situation where it is usually not possible to fulfill all the requirements of the object for its stabilization system. A compromise decision must be accepted. The most striking example in this respect is the case of nonstabilizability [2]], where the object makes conflicting demands of the stabilization system for neighboring frequencies. In this case the work of the stabilization system leads to unstable oscillations on the frequency corresponding to the oscillations of one of the oscillators. In other words, the stabilization system, which is expected to insure stability of movement of the object, cannot handle its assignments and additional design elements, oscillation dampers, must be introducted to disperse the energy of the oscillations which is pumped through the stabilization system. The problems that arise during design of the control system are made more complex when building objects that have multiple purposes and perform a whole set of tasks simultaneously, and also when the absolute dimensions of the object are increased.

When the disturbed motion of such an object is described by a system of differential equations, the number of attached oscillators included in the dynamic scheme increases. The spectrum of frequencies of the system becomes denser and the amplitude-phase frequency characteristic of the stabilization system must meet certain requirements on each of the frequencies corresponding to oscillation of an attached oscillator. It should be observed that the description of the object by a dynamic scheme becomes less reliable as the object becomes more complex and its dimensions increase. This means not only an increase in possible deviations of the actual parameters of the object from those given in the system of equations, but also the fact that the physical premises included in the dynamic scheme of the object may not be entirely correct. The number of attached oscillators included in the dynamic scheme for such a complex object is somewhat indefinite.

The requirements for the phase characteristic of the stabilization system on the frequencies of particular attached oscialltors also may not be adequately substantiated.

The principal goal of the present book is to show that the use of onboard digital computers in the control contour opens up new opportunities for constructing stabilization algorithms of more complex structure, which can be used to support the regulation processes of contemporary space craft.

Let us formulate the basic guidelines to employ in making up the stabilization algorithm.

During construction of the stabilization algorithm the basic principle, which is successive fulfillment of the series of requirements made by the object of its control system, is preserved, as in work [2]. It is unwise to pose the problem of synthesizing an algorithm that optimizes some global criterion, because many different requirements made of the stabilization system must be considered. These conditions differ in nature, reflect highly diverse physical processes, and can be reduced to a general criterion only in artifical terms.

The stabilization algorithm should be synthesized as a set of simpler elements so that to meet a specific condition imposed on the control system all that must be done will be to select the parameters of the corresponding element. It is also essential that the selection of the values of these parameters have minimum impact on solving the other problems that arise during formulation of the object's stabilization system. The multiplicity of tasks which are arising as space technology develops require building space craft of different classes and with different purposes. No single control algorithm can be designed, of course, to solve the problems of control and stabilization for such objects. Nonetheless, the process of designing a stabilization system must be based on the general approach criterion. Greater complexity of the dynamic scheme of the object entails greater complexity in the structure of the stabilization algorithm. When synthesizing stabilization algorithms for complex objects it is necessary to consider this basic point -- as the design becomes more complex, the reliability of the description of the dynamic system decreases. This circumstance reflects an objective trend in the development of space technology and results from the fact that it is necessary to use new design concepts and, consequently, new physical phenomena whose mathematical descriptions have not been adequately developed are possible.

If the effect of introducing new technology is to permit a broadening of the capabilities of the control system, the stabilization algorithm realized on an ODC must take account of the inadequte reliability of the dynamic scheme of the object, and the increased complexity of the structure of the algorithm should be directed to greater use of the elements of identification of the structure and parameters of the object and to introducing elements of adaptation in the algorithm.

There are also design requirements made of the the stabilization algorithm on the ODC. The algorithm that stabilizes the launch vehicles and space craft in the recovery and correction segments with the sustainer engines working must not contain iterative procedures. The algorithm should not include such problems as, for example, determining the roots of the characteristic equation, inverting matrixes, calculating special functions, and the like. It is most acceptable to reduce the stabilization algorithm to finite difference equations.

This statement about the design features of the algorithm does not apply to such specific problems of space craft control as guidance problems, turns around the center of mass, and orientation problems, in other words to those cases where there are significant time intervals between engagement of the engines. For these problems it is often possible to use iterative procedures to solve problems of identifying the dynamic scheme of the object.

Conclusion

In conclusion we will set forth a few principles that can be used during work to synthesize algorithms to control the movement of a space craft. The system to control the angular motion of a multifunctional space craft must have a broad range of tasks, including orientation and stabilization of the object, performance of turns in space, guidance and approach during docking, and numerous others. During an extended space flight there may be changes in the configuration of the object, the composition of measurement and actuating organs, and the mass, moment, strength, and other characteristics of the craft.

Therefore, when constructing particular subsystems that solve particular problems, their interaction in the overal structure of the control system of the space craft must be taken into account. During docking, for example, control of angular motion must be combined with shaping the approach trajectory, whereas the angular stabilization of the launch vehicles depends very little on the characteristics of the trajectory in the active segment. Constructing the control system as a set of essentially autonomous functional units designed to perform particular tasks may lead to significant problems in performing the flight of the space craft.

Including an onboard digital computer in the control system makes it possible to construct the system on the heirarchical principle, because the central processor processes and selects the information coming from lower-level subsystems and also produces and distributes control signals among these subsystems. Considering the many different tasks now being given to the control systems of long-lived space craft, the problem of constructing algorithms that are realizable on ODC's is a central challenge in synthesizing the control system.

The basic requirement made of algorithms to control the movement of space craft is to perform their task with due regard for possible changes in the characteristics of the object and in the composition of measurement and actuating organs. Optimal performance of the given maneuver by the object is desirable, but it should be accomplished so as to fulfill this requirement. The degree of departure from the optimal solution is determined by the skill of the designer.

The method of constructing a stabilization algorithm that has been proposed in this book may be used to synthesize algorithms to control other types of movement by space craft, in particular for problems of guidance and turning the space craft in space. It should be kept in mind that the method of constructing an adaptive algorithm with a model is not a formalized procedure, but rather depends on the essential features of the problem being solved.

The problem of adapting the structure and parameters of the algorithm is solved by analyzing and identifying information that describes the movement of the object. The dynamic scheme corresponding to the angular movement of contemporary space craft is so complex that it is difficult to follow the traditional approach to construction of an adaptive control system based on identification of the structure and parameters of the object's dynamic scheme and then modifying the structure and parameters of the control algorithms. It has been shown in this book, through the example of constructing a stabilization algorithm, that identification involves expanding the observed signal into its constituent parts. The control action is formed on the basis of analysis of each of the components of the observed signal. In this case the algorithm is constructed so that there is no need for detailed study of the question of the causes of oscillating components or obtaining precise quantitative descriptions of them. The appearance of an oscillating component in the observed signal may be the result of flexural oscillations of the body or oscillations of the liquid filler, but this does not affect the response of the adaptive algorithm, which is directed to damping the oscillating component.

The purpose of this work has been to show how the algorithm is shaped, its ability to adapt, and a certain plasticity which is manifested in the process of synthesizing the algorithm. It is assumed here that the degree to which the designer understands the characteristics of the object's dynamic scheme and the designer's understanding of the potential of the algorithm may significantly affect the choice of a model, the selection of the type of filtering elements, and the organization of the computing process in the ODC. Choice of the model is conditioned on the mechanical characteristics of the control problem. The characteristics of the filtering elements, in turn, are determined by the requirements made by the object for amplitude-phase frequency characteristic of the control system and by the ability to reorganize the structure of the algorithm in the adaptation mode. In principl the algorithm can also use more powerful filters, but we should point out that our attempts to use recursive filters and smoothing weighted functions of the Gibbs multiplier type did not prove useful in constructing an algorithm with adaptation. This pointed to the more general principle that any more refined testing tool is more critical of unforeseen changes in the structure and parameters of the control object.

The book has deliberately emphasized the fact that it is not wise to formulate a rigidly regimented method of constructing the control algorithm, similar to the solutions to mathematical problems. The principle of construction of the control algorithm must make it possible to employ the creative capabilities of the builders of the control system, their understanding of the dynamic characteristics of the object and their perception of the constraints and demands made of the control system that have not been formalized in the mathematical statement of the problem.

FOOTNOTES

1. K. A. Abgaryan, and I. M. Rapoport, "Dinamika Raket" [Rocket Dynamics], Moscow, "Mashinostroyeniye," 1969, 376 pages.

2. V. D. Arens, S. M. Fedorov, and M. S. Khitrik, "Dinamika Sistem Upravleniya Raket s Bortovymi Vychislitel'nymi Mashinami" [Dynamics of the Control Systems of Rockets with Onboard Computers], Moscow, "Mashinostroyeniye," 1976, 272 pages.

3. N. N. Bogolyubov, and Yu. A. Mitropol'skiy, "Asimptoticheskiye Metody v Teorii Nelineynykh Kolebaniy" [Asymptotic Methods in the Theory of Nonlinear Oscillations], Moscow, "Fizmatgiz," 1963, 412 pages.

4. "The Onboard Digital Computers of Contemporary Launch Vehicles and Space Craft," VOPROSY RAKETNOY TEKHNIKI, 1970, No 7, pp 3-18.

5. F. R. Gantmakher, "Teoriya Matrits" [Matrix Theory], Moscow, "Nauka," 1967, 576 pages.

6. E. I. Gitis, "Preobrazovateli Informatsii dlya Elektronnykh Tsifrovykh Vychislitel'nykh Ustroystv" [Data Convertors for Electronic Digital Computers], Moscow, "Energiya," 1975, 447 pages.

7. L. Ye. Goncharova, B. I. Rabinovich, and I. M. Sidorov, "Construction of a Control System with a Standard Model," DAN SSSR, 1973, Vol 213, No 5, pp 1037-1039.

8. E. Gumbel', "Statistika Ekstremal'nykh Znacheniy" [The Statistics of Extreme Values], Moscow, "Mir," 1965, 232 pages.

9. R. Kalman, and R. B'yusi, "New Results in Linear Filtration and Prediction Theory," TRUDY AMERIKANSKOGO OBSHCHESTVA INZHENEROV-MEKHANIKOV, SER. D., Vol 33, No 1, IL, 1961.

10. N. I. Kiselev, and I. M. Sidorov, "Statistical Estimate of the Global Extremum," AVTOMATIKA I VYCHISLITEL'NAYA TEKHNIKA, 1974, No 4, pp 45-49.

11. G. Korn, and T. Korn, "Spravochnik po Matematike. Dlya Nauchnykh Rabotnikov i Inzhenerov" [Handbook of Mathematics. For Scientists and Engineers], Moscow, "Nauka," 1977, 832 pages.

12. N. A. Lifshits, V. N. Vinogradov, and G. A. Golubev, "Korrelyatsionnaya Teoriya Optimal'nogo Upravleniya Mnogomernymi Protsessami" [Correlation Theory of Optimal Control of Multidimensional Processes], Moscow, "Sovetskoye radio," 1974, 328 pages.

13. A. I. Lur'ye, "Nekotoryye Nelineynyye Zadachi Teorii Avtomaticheskogo Regulirovaniya" [Some Nonlinear Problems of Automatic Regulation Theory], Moscow-Leningrad, GITTL, 1951, 216 pages.

14. V. G. Lebedev, and I. M. Sidorov, "Algorithm for Turning a Space Craft with Elastic Elements at an Assigned Angle," KOSMICHESKIYE ISSLEDOVANIYA, Vol 12, Vyp 5, 1974, pp 797-799.

15. Dzh. Kh. Lening, and R. G. Bettin, "Sluchaynyye Protsessy v Zadachakh Avtomaticheskogo Upravlaniya" [Random Processes in Automatic Control Problems], Moscow, "Isnotrannaya literatura," 1958, 387 pages.

16. G. N. Mikishev, and B. I. Rabinovich, "Dinamika Tonkostennykh Konstruktsiy s Otsekami, Soderzhashchimi Zhidkost'" [Dynamics of Thin-Walled Design Elements with Compartments Containing Liquids], Moscow, "Mashinostroyeniye," 1971, 563 pages.

17. Ye. P. Popov, "Prikladnaya Teoriya Protsessov Upravleniya v Nelineyniykh Sistemakh" [Applied Theory of Control Processes in Nonlinear Systems], Moscow, "Nauka," 1973, 584 pages.

18. Ye. P. Popov, I. M. Sidorov, and I. P Korotayeva, "Division of Movements by the Harmonic Linearization Technique and Its Use to Synthesize Nonlinear Systems," in "Metody Sinteza Nelineynykh Sistem Avtomaticheskogo Upravleniya" [Methods of Synethsizing Nonlinear Automatic Control Systems], Moscow, "Mashinostroyeniye," 1970, pp 71-86.

19. V. S. Pugachev, "Teoriya Sluchaynkh Funktsiy i Yeye Primeneniye k Zadacham Avtomaticheskogo Upravleniya" [The Theory of Random Functions and Its Application to Problems of Automatic Control], Moscow, "Fizmatgiz," 1960, 652 pages.

20. B. I. Rabinovich, "Study of the Stability of Systems with Multiple Degrees of Freedom," IZVESTIYA AN SSSR, SER. TEKHNICHESKAYA KIBERNETIKA, 1964, No 4, pp 159-169.

21. B. I. Rabinovich, "Vvedeniye v Dinamiku Raket-Nositeley Kosmicheskikh Apparatov" [Introduction to the Dynamics of the Launch Vehicles of Space Craft], Moscow, "Mashinostroyeniye," 1975, 416 pages.

22. A. P. Razygrayev, "Osnovy Upravleniya Poletom Kosmicheskikh Apparatov i Korabley" [Fundamentals of Controlling the Flight of Space Craft and Ships], Moscow, "Mashinostroyeniye," 1977, 472 pages.

23. I. M. Rapoport, "O Nekotorykh Asimptoticheskikh Metodakh v Teorii Differentsial'nykh Uravneniy" [Some Asymptotic Methods in the Theory of Differential Equations], Moscow, Izd. AN SSSR, 1954, 287 pages.

24. B. V. Raushenbakh, and Ye. N. Tokar', "Upravleniye Oriyentatsiyey Kosmicheskikh Apparatov" [Control of the Orientation of Space Craft], Moscow, "Nauka," 1974, 600 pages.

25. V. M. Rogovoy, and S. V. Cheremnykh, "Dinamicheskaya Ustoychivost' Kosmicheskikh Apparatov s ZhRD" [Dynamic Stability of Space Craft with Liquid-Fuel Rocket Engines], Moscow, "Mashinostroyeniye," 1975, 152 pages.

26. I. M. Sidorov, and N. N. Balashova, "Control of the Movement of a Space Craft Entering the Atmosphere at Escape Velocity," KOSMICHESKIYE ISSLEDOVANIYA, 1973, Vol 11, No 3, pp 388-396.

27. I. M. Sidorov, and L. Ye. Goncharova, "The Stability of Nonstationary Control Systems with Nonlinear Correction Devices," in "Problemy Navigatsii i Avtomaticheskogo Upravleniya" [Problems of Navigation and Automatic Control], Izd. VINITI, 1969, Vyp 1, pp 45-52.

28. I. M. Sidorov, L. Ye. Goncharova, and V. I. Prokhorenko, "Recognition of the Spectral Structure of Control Processes," in ibid., pp 53-59.

29. I. M. Sidorov, and I. P. Korotayeva, "Study of Nonlinear Systems in the Case of Establishing Two-Frequency Processes," IZV. AN SSSR. SER. TEKHNICHESKAYA KIBERNETIKA, 1969, No 5, pp 148-158.

30. I. M. Sidorov, and I. P. Korotayeva, "Study of the Structural Stability of Mechanical Systems with Many Degrees of Freedom in the Presence of a Correcting Device," IZV. AN SSSR. SER. TEKHNICHESKAYA KIBERNETIKA, 1965, No 5, pp 183-187.

31. I. M. Sidorov, L. B. Krangacheva, and V. G. Lebedev, "Construction of a Stabilization Algorithm for a Deformable Space Craft Using an Onboard Digital Computer," KOSMICHESKIYE ISSLEDOVANIYA, 1973, Vol 11, Vyp 3, pp 388-396.

32. I. M. Sidorov, and V. V. Timofeyev, "Analysis of Two-Frequency Regimes of Oscillations in a Stabilization System," IZV. AN SSSR. SER. TEKHNICHESKAYA KIBERNETIKA, 1978, No 1, pp 174-182.

33. I. M. Sidorov, and S. V. Cheremnykh, "One Method of Studying the Stability of Regulable Systems," INZH. ZHURNAL. MEKHANIKA TVERDOGO TELA, 1967, No 2, pp 74-80.

34. N. V. Smirnov, and I. V. Dunin-Barkovskiy, "Kurs Teorii Veroyatnostey i Matematicheskoy Statistiki dlya Tekhnicheskikh Prilozheniy" [Course in Probability Theory and Mathematical Statistics for Engineering Applications], Moscow, "Nauka," 1965, 347 pages.

35. A. V. Solodov, "Metody Teorii Optimal'nykh Sistem v Zadache Nepreryvnoy Lineynoy Fil'tratsii" [Methods of the Theory of Optimal Systems in the Problem of Continuous Linear Filtration], Moscow, "Nauka," 1976, 252 pages.

36. K. Spidi, R. Braun, and Dzh. Gudvin, "Teoriya Upravleniya. Identifikatsiya i Optimizatsiya Sistem Upravleniya" [Control Theory. Identification and Optimization of Control Systems], Moscow, "Mir," 1973, 214 pages.

37. Yu. T. Tu, "Tsifrovyye i Impul'snyye Sistemy Avtomaticheskogo Upravleniya" [Digital and Pulsed Automatic Control Systems], Moscow, "Mashinostroyeniye," 1964, 703 pages.

38. Firmen and Tite, "Digital Stabilization System of the Titan 3S Launch Vehicle," VOPROSY RAKETNOY TEKHNIKI, 1970, No 7, pp 42-58.

39. Yu. A. Tsurikov, "The Stability of One Dynamic System," INZH. ZHURNAL. MEKHANIKA TVERDOGO TELA, 1966, No 2, pp 193-195.

40. Ya. Ye. Tsypkin, "Teoriya Lineynykh Impul'snykh Sistem" [Theory of Linear Pulsed Systems], Moscow, "Nauka," 1963, 968 pages.

41. F. Chaki, "Sovremennaya Teoriya Upravleniya" [Contemporary Control Theory], Moscow, "Mir," 1975, 376 pages.

42. V. A. Shatalov, S. N. Seletkov, and B. S. Skrebushevskiy, "Primeneniye EVM v Sisteme Upravleniya Kosmicheskimi Apparatami" [The Application of Computers in Systems to Control Space Craft], Moscow, "Mashinostroyeniye," 1972, 240 pages.

Table of Contents

11,176
CSO: 1863/50

UDC 519.86: 681.3

MACHINE MATHEMATICAL MODELING

Moscow REALIZATSIYA MATEMATICHESKIKH MODELEY NA EVM in Russian 1981 (signed to press 22 Jun 81) pp 2-5, 143-144, 168-174

[Annotation, foreword, excerpt from chapter 6.1, bibliography and table of contents from book "Machine Realization of Mathematical Models" by Viktor Aleksandrovich Leont'yev, Izdatel'stvo "Energiya", 7,000 copies, 175 pages]

[Excerpts] Annotation.

The problem of controlling a set of programs based on analysis of a real-time "demand-production" model is examined. The criteria for effectiveness selected are those associated with model dimension, accuracy in modeling and the amount of nonproductive use of machine time. Effective algorithms are derived for solving allocation and traveling salesman problems.

The book is intended for engineering and technical personnel engaged in the development of models and algorithms, and also for statistical analysis of economic efficiency in automated control systems.

Foreword.

The further upsurge in the national economy and the more complete satisfaction of the public's material and spiritual needs can be accomplished primarily through growth in the efficiency of social production and the acceleration of the scientific and technical progress [1,2].

A special place is now assigned to improvements in labor productivity, the rational utilization of material resources and manpower, and economical opening up and exploitation of natural resources. And in this business, an exceptionally important part is being played by improvements in the planning mechanism and the organizational forms of management, and also in economic accounting based on the development of state and sector automated control systems and special mathematical models for solving the most urgent and complex socioeconomic problems.

The fundamental principle in building control systems and large models is the systems approach whose essence has been described in [3, page 455]:

1) formulating goals and clarifying their hierarchy before the initiation of any activity associated with control; this includes decisionmaking.

2) obtaining maximum effect in the sense of achieving the goals set with minimum cost by comparative analysis of alternative pathways and methods of achieving the goals and making the appropriate choice;

3) quantitative analysis (quantification) of goals and the methods and facilities used to achieve them, based on a broad and comprehensive evaluation of all possible and planned results of activity.

Thus, the systems approach is a generalized, organizational-management principle characterizing the highest stage in the development of the control process over a given class of objects and it arms the developer and researcher with a powerful, standardized tool for analyzing and synthesizing systems that differ in terms of functions, character and structure and gives them a scientifically sound "template" for action. In accordance with this approach, during the first stage individual optimization problems and sets are resolved, special mathematical methods and approaches are worked out for problem solving and methods for analysis and measurement of the status of objects, and particular control problems are studied, and so forth.

It is with the solving of precisely these questions, and also with the methods for realizing and computing operating conditions for machine models of complex processes, that this book deals.

The basic problems selected by the author are selection of dimension for the objects modeled, accuracy in modeling and minimization of nonproductive use of machine time. In accordance with this, methods and pathways have been worked out for achieving set goals: an algorithm for selecting model dimension; a method for metareductions [metod metaprivedeniy] for solving allocation problems, to which the problem of modeling accuracy is reduced; algorithms for solving traveling salesman problems in enhanced dimension, in which a rational sequencing organization is set for the operation of units in the program set.

It should be noted that the need to solve the problem of model dimension using machine methods has now come to a head; but work devoted to this question can be found neither in the Soviet nor the foreign literature. The method of metareductions, based on iterative maximization of dual evaluation of allocations within a problem as a problem in linear programing with specific matrix constraints may be of interest to specialists in the the theory of algorithms and computing methods. In a number of instances, the features of this method have required the introduction of nontraditional terminology.

The approaches developed to controlling the realization of program sets are oriented largely on the application of economic dynamics in the models. However, problems of selecting model dimension and minimizing nonproductive use of machine facilities can be applied to a broad class of models. The method of metareductions in discrete programing presented is not linked with the features of any particular models and can be used in all cases where the control problem for the process being considered is reduced to a problem of allocations.

The author expresses his deep gratitude to doctors of technical sciences Yu.M. Fatkin and I.V. Goryachev, candidates of physicomathematical sciences V.A. Antipov, V.L. Arlazarov and Ye.A. Dinits, and candidates of technical sciences A.L. Uzdemir and I.I. Bronshteyn, who participated at various stages in examining and discussing the material forming the basis of the book; and also to T.N. Makarova and T.S. Rukina for their help in preparing the material for print.

The author expresses his special thanks to candidate of technical sciences V.N. Buslenko for his important criticisms and advice, which have been taken into account in the final version of the book.

Enquiries and recommendations should be sent to: 113114, Moscow, M-114, Shlyuznaya naberezhnaya 10, "Energiya" Publishing House.

Chapter 6. Technical Means for Automating the Acquisition of Statistical Data.

In this chapter we consider special devices that free the machine user from the need to carry out laborious, nonproductive work on acquisition, logging and initial processing of statistical input data (that is, determination of matrix distances), and conduct certain procedures that have traditionally been done directly in the computer. Thus, in the devices described, a number of algorithms are derived for initial data processing, leading to a reduction in the volume of input data and thus to a reduction in computing operations. In other cases the devices are designed to carry out operations following which the search for optimal plans is accelerated.

The devices described can be used in control of industrial output production, for example, filler (for drawing synthetic fibers), for analyzing specimens in studies of metallic structures and geological samples, for doing electron microscope studies on the quality of stored foodstuffs, and also in all cases where a large number of small-dimension objects in a study have heating temperatures, degrees of illumination, color and other attributes that differ sharply from the levels in the background against which the objects are placed.

From a design viewpoint, the devices are a set of standard electronic components, which execute measurement and logic functions and which are controlled with the aid of a (control) program unit and memory. The unit should be thought of as either a general-purpose or a control computer.

BIBLIOGRAPHY

1. "Materialy XXVI s'yezda KPSS" [Materials on the 26th CPSU Congress], Moscow Politizdat, 1981, p 223.

2. Brezhnev, L.I. "Velikiy Oktyabr i progress chelovechestva. Doklad na sovmestnom torzhestvennom zasedanii Tsentral'nogo Komiteta KPSS, Verkhovnogo Soveta SSSR i Verkhovnogo Soveta RSFSR v Kremlevskom Dvortse s'yezdov 2 noyabrya 1977 goda" [The Great October and Human Progress. Reported Delivered at a Joint Ceremonial Session of the CPSU Central Committee, USSR Supreme Soviet and RSFSR Supreme Soviet in the Kremlin Palace of Congresses, 2 November 1977], Moscow, Politizdat, 1977, p 30.

3. Gvishiani, D.M. "Organizatsiya i upravleniye" [Organization and Control], Moscow, Nauka, 1972, p 536.

4. "Metodologicheskiye voprosy postroyeniya imitatsionnykh sistem: Obzor/S.V. Yemel'yanov, V.V. Kalashnikov, V.I. Lutkov, B.V. Nemchirov" [Methodological Questions in Building Simulation Systems. A Collection of Works by S.V. Yemel'yanov, V.V. Kalashnikov, V.I. Lutkov and B.V. Nemchirov.], edited by D.M. Gvishiani and S.V. Yemel'yanov, Moscow, International Center for Scientific and Technical Information, 1978, p 88.

5. "Ekspertnyye metody v sistemnykh issledovaniyakh" [Evaluation Methods in Systems Research] edited by USSR Academy of Sciences academician D.M. Gvishiani and corresponding member of the USSR Academy of Sciences S.V. Yemel'yanov, in "Sbornik trudov VNII sistemnykh issledovaniy" [Collection of Works by the All-Union Scientific Research Institute for Systems Research], Moscow, USSR State Committee for Science and Technology, 1979, 4th edition, p 88.

6. Yu.M. Fatkin, L.N. Blyoskina, J.N. Dmitrieva et al. "Hierachical imitation demand-supply model. Preprints IFAK 75, Sixth Triennial World Congress, Boston/Cambridge, Massachisetts, pt III D, session 62; Urban, Regional and National Planning, 62.5.

7. Avanesov, Yu.A. "Prognozirovaniye sprosa v roznichnoy torgovle" [Predicting Demand in the Retail Trade], Moscow, Ekonomika, 1975, p 103.

8. Likhtenshteyn, V.Ye. "Modeli diskretnogo programmirovaniya" [Models in Discrete Programing], Moscow, Nauka, 1971, p 328.

9. Balas, E. "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," OPERATIONS RES., 1965, Vol 13, No 4, pp 517-546.

10. Semenova, V.A. "Opredeleniye ratsional'nogo varianta spetsializatsii sel'skokhozyaystvennykh predpriyatiy v rayone" [Determination of the Rational Variant in Rayon Specialization of Agricultural Enterprises] in "Ekonomiko-matematicheskiye metody planirovaniya i analiza sel'skokhozyaystvennogo proizvodstva" [Economic-Mathematical Methods in Planning and Analysis of Agricultural Production], Novosibirsk, Nauka (Siberian Branch), 1969, pp 93-115.

11. Granberg, A.G. "Sel'skoye khozyaystvo v modeli optimal'nogo territorial'no-proizvodstvennogo kompleksa" [Agriculture in a Model for the Optimal Territorial-Production Complex] in "Economic-Mathematical Methods in Planning and Analysis of AgriculturalProduction," Novosibirsk, Nauka (Siberian Branch), 1969, pp 53-73.

12. Dudkin, L.M. and Vakhutinskiy, I.Ya. "Vzaimouvyazka optimal'nykh planov raznykh urovney upravleniya metodami iterativnogo agregirovaniya" [Coordination of Optimal Plans at Different Control Levels Using the Methods of Iterative Sequencing] Moscow, Nauka, 1975, p 256.

13. Dinits, Ye.A. "Algoritm porazryadnogo sokrashcheniya nevyazok i transportnyye zadachi" [An Algorithm for Bit Serial Contraction of Discrepancies and Transport Problems] in "Issledovaniya po diskretnoy matematike" [Studies in Discrete Mathematics], Moscow, Nauka, 1973, pp 46-57.

14. Mikhalevskiy, B.N. "Model' narodnogo khozyaystva" [A Model of the National Economy] in "Slovar'-spravochnik: Matematika i kibernetika v ekonomike" [A Handbook of Mathematics and Cybernetics in Economics], Moscow, Ekonomika, 1971, p 97.

15. Dinits, Ye.A. and Kronrod, M.A. "Odin algoritm resheniya zadachi o naznachenii" [One Algorithm for Solving the Allocation Problem], DOKLADY AN SSSR, 1969, Vol 189, No 1, pp 23-25.

16. Entsiklopediya kibernetiki [Encyclopedia of Cybernetics], Vols 1-2, Kiev, Edited by the Ural'sk Soviet Encyclopedia, 1975. p 374.

17. Bellman, R. and Endzhel, E. "Dinamicheskoye programmirovaniye i upravleniya v chastnykh proizvodnykh" [Dynamic Programing and Control in Partial Derivatives], Moscow, Mir, 1974, p 207.

18. Chernous'ko, F.L. and Banichuk, N.V. "Variatsionnyye zadachi mekhaniki i upravleniya" [Variation Problems in Mechanics and Control], Moscow, Nauka, 1973, p 238.

19. Korbut, A.A. and Finkel'shteyn, Yu.Yu. "Diskretnoye programmirovaniye" [Discrete Programing], Moscow, Nauka, 1972, p 368.

20. Kantorovich, L.V. and Gorstko, A.B. "Optimal'nyye resheniya v ekonomike" [Optimal Decisions in Economics], Moscow, Nauka, 1972, p 231.

21. Zangvili, U.I. "Nelineynoye programmirovaniye" [Nonlinear Programing], Moscow, Sovetskoye Radio, 1973, p 311.

22. Yudin, D.B. and Gol'shteyn, Ye.G. "Zadachi i metody lineynogo programmirovaniya" [Problems and Methods in Linear Programing], Moscow, Sovetskoye Radio, 1964. p 736.

23. "Statisticheskoye izucheniye sprosa i potrebleniya" [Statistical Studies of Demand and Consumption], Moscow, Nauka, 1966, p 259.

24. Lur'ye, A.L. "Problemy optimizatsii ekonomicheskoy sistemy" [Problems in Optimizing the Economic System], Moscow, Moscow State University, 1969, pp 3-25.

25. Kossov, V.V. "Mezhotraslevoy balanc" [The Intersector Balance]. Moscow, Ekonomika, 1966, p 223.

26. Ivanov, V.V. "Voprosy tochnosti i effektivnosti vychislitel'nykh algoritmov (Obzor dostizheniy v oblasti kibernetiki i vychislitel'noy tekhniki)" [Questions of Accuracy and Efficiency in Computing Algorithms. A Review of Achievements in the Field of Cybernetics and Computer Technology], Kiev, Ukrainian SSR Publishing Commission, 1969, p 135.

27. Sergiyenko, I.V. "Metody organizatsii vychislitel'nogo protsessa na vychislitel'nykh mashinakh. Obzor" [Methods of Organizing the Computing Process in Computers. A Review], Kiev, Ukrainian SSR Publishing Commission, 1971, p 162.

28. Lipayev, V.V. and Yashkov, S.F. "Effektivnost' metodov organizatsii vychislitel'nogo protsessa v ASU" [Efficiency in Methods for Organizing the Computing Process in Automated Control Systems], Moscow, Statistika, 1975, p 255.

29. Papkov, V.I. "O statisticheskom issledovanii programm bystrodeystvuyushchikh vychislitel'nykh mashin" [Statistical Studies on Programs in High-Speed Computers], in "Vychislitel'naya tekhnika" [Computer Equipment], Leningrad, Energiya, 1972, Second edition, pp 61-67.

30. IVANOV, V.V. "Nekotoryye voprosy optimizatsii vychisleniy" [Questions of Computing Optimization] in "Matematicheskoye obespecheniye ETsVM" [Software for Digital Computers], Kiev, Ukrainian SSR Publishing Commission, 1972, pp 149-172.

31. Kavalerov, G.I. and Mandel'shtam, S.M. "Vvedeniye v informatsionnuyu teoriyu izmereniy" [Entry in Information Theory of Measurements], Moscow, Energiya, 1974, p 375.

32. Rozenberg, V.Ya. "VVedeniye v teoriyu tochnosti izmeritel'nykh sistem" [Entry in the Theory of Accuracy in Measuring Systems], Moscow, Sovetskoye Radio, 1975, p 304.

33. Sorenkov, E.I.; Teliga, A.I. and Shatalov, A.S. "Tochnost' vychislitel'nykh ustroistv i algoritmov" [Accuracy in Computing Devices and Algorithms], Moscow, Mashinstroyeniye, 1976, p 200.

34. Venetskiy, I.G. and Kil'dishev, G.S. "Teoriya veroyatnostey i matematicheskaya statistika" [Probability Theory and Mathematical Statistics], Moscow, Statistika, 1975, p 264.

35. Gmurman, V.Ye. "Rukovodstvo k resheniyu zadach po teorii veroyatnostey i matematicheskoy statistike" [Handbook for Solving Problems in Probability Theory and Mathematical Statistics], Moscow, Vysshaya shkola, 1975, p 333.

36. Demidovich, B.P.; Maron, I.A. and Shuvalova, E.Z. "Chislennyye metody analiza" [Numerical Analytical Methods], Moscow, State Publishing House of Physics and Mathematical Literature, 1963, p 400.

37. Yudin, D.B. and Gol'shteyn, Ye.G. "Lineynoye programmirovaniye" [Linear Programing], Moscow, Nauka, 1969, p 424.

38. Gol'shteyn, Ye.G. and Yudin, D.B. "Zadachi lineynogo programmirovaniya transportnogo tipa" [Transportation-Type Problems in Linear Programing], Moscow, Nauka, 1969, p 384.

39. Kun, G. "Vengerskiy metod resheniya zadachi o naznacheniy" [The Hungarian Method for Solving the Allocation Problem] in "Metody i algoritmy resheniya transportnoy zadachi" [Methods and algorithms for Solving the Transportaion Problem], Moscow, Gosstatizdat, 1963, pp 92-107.

40. Florian, M. and Morton, M. An Experimental Evaluation of Some Methods of Solving the Assignment Problem." CANADIAN OPERATIONAL RESEARCH SOCIETY, Ottawa, 1970 Vol 8, No 2, pp 101-108.

41. Gavett, T.W. and Plyter, Norman V. "The Optimal Assignment of Facilities to Locations by Branch and Bound." OPERATIONS RESEARCH, 1966, Vol 14, No 2, pp 210-232.

42. Klein, M. "A Primal Method for Minimal Flow Costs with Applications to the Assignment and Transportation Problems" MANAGEMENT SCIENCE, 1967, Vol 14, No 3, pp 205-220.

43. Garfinkel, R.S. "An Improved Algorithm for the Bottonesk [sic] Assignment Problem." OPERATIONS RESEARCH, 1971, Vol 19, No 7, pp 1747-1751.

44. Leont'yev, V.K. "Algebraicheskaya struktura nekotorykh zadach diskretnogo programmirovaniya" [Algebraic Structure of Some Problems in Discrete Programing], in "Problemy Kibernetiki" [Problems of Cybernetics], 26th edition, Moscow, Nauka, 1973, pp 277-290.

45. Little, Dzh.; Murti, K.; Suini, D. and Kerel, K. "Algoritm dlya resheniya zadachi o kommivoyazhere" [An Algorithm for Solving the Traveling Salesman Problem], EKONOMIKA I MATEMATICHESKIYE METODY, 1965, No 1, pp 94-107.

46. Leont'yev, V.A. "Metod metaprivedeniy dlya zadachi o naznacheniyakh. Tezisy dokladov na VI Vsesoyuznoy konferentsii po ekstremal'nym zadachami" [A Method of Metareductions for Allocations Problem. Theses of Reports at the 6th All-Union Conference on Extremal Problems], Part II, Tallin, Estonian SSR Academy of Sciences Publishing Commission, 1973, pp 17-18.

47. Leont'yev, V.A. "Metod metaprivedeniy k resheniyu nekotorykh zadach kombinatornogo tipa" [A Metareduction Method for Solving Certain Problems of the Combinatorial Type], KIBERNETIKA, 1974, No 7, Abstract 7G621, p 69.

48. Ore, O. Teoriya grafov" [Graph Theory], Moscow, Nauka, 1968, p 352.

49. Popecku, Dorina Comsa. "An Algorithm for Solving the Transportation Problem. Economic Computation and Economic Cybernetics Studies and Research." Bucharest, 1976, No 2, pp 41-51.

50. Verkhovskiy, B.S. "Simmetrichnyye mnogoindeksnyye transportnyye zadachi" [Symmetric Multiple Index Transportation Problems] in "Sbornik: Problemy optimal'nogo planirovaniya, proyektirovaniya i upravleniya proizvodstvom" [Collection of Works: Problems in Optimal Planning, Design and Control in Production], Transactions of a Moscow State University Theoretical Conference, Moscow, Moscow State University, 1963, pp 482-498.

51. Khutsishvili, I.G. "Ob odnoy zadache razmeshcheniya" [An Allocation Problem], Sakartvelos SSR Metsniyerebata Akademiks moambe. Reports of the Georgian SSR Academy of Sciences, 1971, Vol 63, No 3, pp 557-559.

52. Anisimov-Spiridonov, D.D. "Metod lineynykh vetvleniy i nekotoryye ego primeneniya v optimal'nom planirovanii" [A Method of Linear Branching and Some of its Applications in Optimal Planning], Moscow, Nauka, 1964, p 119.

53. Anisimov-Spiridonov, D.D. "Metody i modeli bol'shikh sistem optimal'nogo planirovaniya i upravleniya" [Methods and Models for Large Systems in Optimal Planning and Control], Moscow, Nauka, 1969, p 360.

54. Gol'shteyn, Ye.G. "Transportnaya zadacha i yeye obobshcheniya" [The Transportation Problem and its Generalization] in "Sbornik: Metody i algoritmy resheniya transportnoy zadachi" [Collection of Works: Methods and Algorithms for Solving the Transportation Problem], Moscow, Gosstatizdat, 1963, pp 3-34.

55. Dualer, P. and Galler, B. "Primeneniye metoda privedennykh matrits k resheniyu obshchey transportnoy zadachi" [Application of the Reduced Matrix Method in Solving the General Transportation Problem] in "Methods and Algorithms for Solving the Transportation Problem," Moscow, Gosstatizdat, 1963, pp 115-121.

56. Revenko, V.L. and Oleyarsh, G.B. "A Problem in Allocation," KIBERNETIKA, 1972, No 2, pp 69-70.

57. Yefimov, A.I. and Maksimilian, S.V. "A Four-Index Transportation Problem" in "Matematicheskiye metody v ekonomike" [Mathematical Methods in Economics], Kishinev, Shtinitsa, 1971, 3d edition, pp 3-34.

58. Staykova-Penkova, N. "Trekhindeksna transportna zadacha" [A Three-Index Transportation Problem], Transactions of the Higher Economics Institute, Sofia, I, 1970 (1971), pp 1-19.

59. Pierskalla, W.P. "The Multidimensional Assignment Problem," OPERATIONS RESEARCH, 1968, Vol 16, No 2, pp 422-431.

60. Haley, K.B. "The Solid Transportation Problem," OPERATIONS RESEARCH, 1962, Vol 10, No 4, pp 448-463.

61. Corban, A. "Un model multidimensional de transport" [A Model for Multi-dimensional Transportation], Studii si cercetari matematice, Bucharest, 1972, 24, No 7, pp 1019-1082.

62. Junginger, W. "Ueber die Loesung des dreidimensionalen Transportproblems" [Solving the Three-Dimensional Transportation Problem], Diss., Dokt. Naturwiss, Univ. Stuttgart, 1970.

63. Leue. C. "Methoden zu Ordnungsproblemen" [Methods in Sequencing Problems], Angewandte Information, Braunschweig, 1972, Vol 14, No 4, pp 154-162.

64. Miller, C.E.; Tucker, A.W. and Zemlin, R.A. "Integer Programming Formulation of Traveling Salesman Problem," J. ASSOCIATER FOR COMPUTING MACHINES, 1960, Vol 7, No 4, pp 326-329.

65. Bellmore, M. and Nemhauser, G.L. "The Traveling Problem [sic]: a Survey." OPERATIONS RESEARCH, 1968, Vol 16, No 3, pp 538-558.

66. Burkov, V.N. and Lovetskiy, S.Ye. "Methods for Solving Extremal Problems of the Combinatorial Type (a Review)" AVTOMATIKA I TELEMEKHANIKA, 1968, No 11, pp 69-93.

67. Lawler, E.L. and Wood, D.E. Branch-and-Bound Methods: a Survey," OPERATIONS RESEARCH, 1966, Vol 14, No 4, pp 699-719.

68. Bellman, R. "Primeneniye dinamicheskogo programmirovaniya k zadache o kommivoyazher" [Application of Dynamic Programming in the Traveling Salesman Problem], in Kiberneticheskiy sbornik" [Collection of Works on Cybernetics], Moscow, Mir, 1964, pp 219-222.

69. Lin, S. "Computer Solution of the Traveling Salesman Problem," BELL SYSTEM TECHNICAL JOURNAL, 1965, Vol 44, No 10, pp 2245-2269.

70. Webb, M.H.J. "Some Problems of Producing Approximate Solutions to Traveling Salesman Problem with Hundreds of Thousands of Cities," OPERATIONAL RESEARCH QUARTERLY, 1971, Vol 22, No 1, pp 49-66.

71. Eastman, W.L. "Linear Programming with Pattern Constraints," Ph. D. dissertation, Harvard, 1958.

72. Shapiro, D. "Algorithms for the Solutions of Optimal Cost Traveling Salesman Problem," Sc. D. Thesis, Washington University, St. Louis, 1966.

73. Bellmore, M. and Malone, J.C. "Pathology of the Traveling Salesman Subtour-Elimination Algorithms," OPERATIONS RESEARCH, 1971, Vol 19, No 2, pp 278-307.

74. "Sbornik: Metody i algoritmy resheniya transportnoy zadachi" [Collection of Works: Methods and Algorithms for Solving the Transportation Problem], Moscow, Gosstatizdat, 1963, p 151.

75. Kofman, A. and For, R. "Zaymemsya issledovaniyem operatsiy" [Operations Studies], Moscow, Mir, 1966, p 279.

76. Barachet, L.L. "Graphic Solution of the Traveling Salesman Problem," OPERATIONS RESEARCH, 1957, Vol 5, No 6, pp 841-845.

77. Karg, L.L. and Thompson, G.L. "A Heuristic Approach to Solving Traveling Salesman Problems," MANAGEMENT SCIENCE, 1963-1964, Vol 10, No 2, pp 225-248.

78. Leont'yev, V.A. "Postroyeniye na zadannom mnozhestve tochek gamil'tonova tsikla, blizkogo po dline k naukratchayshemu" [Building on a Given Number of Points in a Hamiltonian Cycle Close to the Shortest Length] in "Aktual'nyye voprosy tekhnicheskoy kibernetiki" [Urgent Problems of Cybernetics Equipment], Moscow, Nauka, 1972, pp 244-248.

79. Author's Certificate No 331406 (USSR). A Computing Device for Solving the Symmetric Traveling Salesman Problem/V.A. Leont'yev, Published in BYULLETEN IZOBRETENIY, 1972, No 9.

80. Author's Certificate No 385279 (USSR). A Device for Solving the Symmetric Traveling Salesman Problem/V.A. Leont'yev. Published in BYULLETEN IZOBRETENIY, 1973, No 25.

81. Kheld, M. and Karp, R.M. "Primeneniye dinamicheskogo programmirovaniya k zadacham uporyadocheniya" [Application of Dynamic Programing in Ordering Problems] in "Kiberneticheskiy sbornik" [A Collection of Works on Cybernetics], Moscow, Mir, 1964, No 9, pp 202-218.

82. Author's Certificate No 397915 (USSR). A Device for Sampling Oblique Arcs on a Graph/V.A. Leont'yev. Published in BYULLETEN IZOBRETENIY, 1973, No 37.

83. Author's Certificate No 432550 (USSR). A Device for Scanning Two-Dimensional Parameter Fields/V.A. Leont'yev. Published in BYULLETEN IZOBRETENIY, 1974, No 22.

84. Author's Certificate No 427351 (USSR). A Device for Analyzing Point Source Images/V.A. Leont'yev. Published in BYULLETEN IZOBRETENIY, 1974, No 17.

85. Author's Certficate No 402735 (USSR). A Method for Determining Distances Between Objects/V.A. Leont'yev. Published in BYULLETEN IZOBRETENIY, 1973, No 42.

Contents

9642
CSO: 1863/53

UDC 658.512.2

AUTOMATION OF EXPLORATORY DESIGN (ARTIFICIAL INTELLIGENCE IN MACHINE DESIGN)

Moscow AVTOMATIZATSIYA POISKOVOGO KONSTRUIROVANIYA (ISKUSSTVENNYY INTELLEKT V MASHINNOM PROYEKTIROVANII) in Russian 1981 (signed to press 18 May 81) pp 2-5, 298-302

[Annotation, foreword, conclusion of book "Automation of Exploratory Design (Artificial Intelligence in Machine Design)" by Aleksandr Ivanovich Polovinkin, Nikolay Konstantinovich Bobkov, Genrikh Yazepovich Bush, Valentin Georgiyevich Grudachev, Aleksandr Mikhaylovich Dvoryankin, Sergey Andreyevich Kudryavtsev, Petr Matveyevich Mazurkin, Vasiliy Vasil'yevich Merkur'yev, Mark Abramovich Moldavskiy, Sergey Arnol'dovich Nikolayev, Gennadiy Sergeyevich Oshchepkov, Emma Pavlovna Sarkisova, Oleg Ignat'yevich Semenkov, Anatoliy Nikolayevich Sobolev, Yevgeniy Arsen'yevich Smirenskiy and Yuliy Tsezarevich Faytel'son, Izdatel'stvo "Radio i svyaz'", 10,000 copies, 344 pages]

[Text]

Annotation

The new area of automation of exploratory design is explained, with primary attention devoted to heuristic and machine methods for finding new technical treatments.

The book is intended for engineers involved in developing and applying methods of finding new design treatments, as well as other specialists.

Foreword

Intensive research and development has recently been underway for the creation and practical application of systems for automating planning, design and technological production preparation for various classes of technical devices. The primary purpose of these systems is to reduce the amount of time and labor required to develop articles and, most importantly, to improve their quality.

The goal of practically all development work is to create and produce articles on a level with the best models in the world. However, this goal is most often not achieved, since obtaining such articles requires inventing a complex of coordinated, as well as highly efficient, new technical treatments. This requires synthesizing and analyzing numerous treatment versions, which for many reasons is difficult or impossible without using computers. At the same time, formalizing and programming the process of searching for (synthesizing) new technical treatments involves the statement and resolution of problems of technical creativity, and causes significant difficulties.

The first automatic design systems made it possible to automate all stages and operations (making engineering analyses, finding required data, optimizing parameters of assigned technical treatments, comparing drawings and other technological planning documentation, testing experimental models, etc.), except for selecting improved and new planning and design treatments. Only recently have convincing results been obtained both here and abroad which make it possible to resolve the problem of automating exploratory design, or (according to the foreign literature) to use artificial intelligence in machine design.

We can say that, on the basis of results which have been achieved in automating exploratory design, development has begun on second-generation automatic design systems with improved creative potential. There is justification to assert that in the near future only second-generation automatic design systems which include exploratory design subsystems will in most cases make it possible to create new articles on a level with the world's best models. The automated design system is one of the main and most important areas in which exploratory design methods are used. However, as will be shown below, this is not its only sphere of application.

This book disseminates the long years of results achieved by the collective of authors, who have been developing the problem of automating exploratory design. The first three sections of the book explain individual, and we might say universal, methods which can be used independently from one another (or in various combinations) to solve various exploratory design problems in the area of instrument building, machine building and construction. Part 1 presents heuristic methods aimed at applications not involving computers; the possibility of using computers is indicated at the same time. Sections 2 and 3 present special machine methods whose use without computers is inadvisable or impossible. Section 4 presents the conception of the creation of exploratory design subsystems for second generation automated design systems.

The present book is not primarily a scientific monograph, but rather a practical aid for developing and applying exploratory design methods and systems. Besides specialists in the development and utilization of automated design systems, this aid will be of interest for engineers in all specialities, and accordingly to students at technical training schools who are interested in methods of finding new design treatments, and for inventors, patent holders and other specialists who are interested in problems of technical creativity and artificial intelligence. In this connection, instead of providing theoretical and experimental foundation for the methods, the book only provides references to the appropriate literature; brief bibliographic references are given rather than a substantive review and analysis of the work of other authors; because of limited space, only the basic construction ideas are given for many algorithms, with references to the literature for more detailed explanation; many examples are limited to the initial data and end results, omitting the intermediate derivations.

The problem in question has required integrated coverage of a broad group of interrelated problems, which resulted in definite difficulties in the exposition. Considering that this is the first attempt to write such a book, the authors request to be forgiven in advance if some places are not entirely successful, and will be glad to receive any comments and remarks to improve the book.

The authors are extremely grateful to Academicians V.M.Glushkov, G.I. Marchuk, I.F. Obraztsov and B.N. Petrov, to Ukrainian SSR Academician V.S. Mikhalevich, to USSR Academy of Sciences Corresponding Members M.A. Gavrilov and G.S. Pospelov, to G.P. Sofonov, President of the Central Committee of the All-Union Society of Inventors and Rationalizers, to professors Yu. B. Borodulin, B.F. Goryunov, N.G. Zagoruyko, G.P. Zakharov, E.K. Kalinin, Yu.V. Kapitonova, I. Myuller, V.A. Myasnikov, A.I. Petrenko and D.A. Pospelov, as well as to engineer Yu.F. Morozov for their valuable critiques and for supporting research on the problem, which undoubtedly helped to improve the manuscript. The authors thank the leadership of the Mariyskaya CPSU oblast committee and the Yoshkar-Ola CPSU city committee, primarily comrades V.P. Nikonov and G.N. Vodovatov for their years of assistance, which facilitated formation of the collective and supported its fruitful work. The authors also express their gratitude to official reviewers Professor O.N. Trifonov and Docent A.Ya. Medvedev, who made a number of valuable comments on the manuscript which helped to improve portions of the book significantly.

Conclusion

This book presents three heuristic methods (Chapters 1-3) which have been tested in practice and which most designers can use without computers to solve many exploratory design problems. It is indicated simultaneously that these methods can be formalized and programmed in most cases, thus significantly increasing their efficiency. Special notice should be given the generalized heuristic method (Chapters 2,3) which can be used as the basis for obtaining a large number of modifications of specialized heuristic methods aimed at individual classes of technical systems and (or) types of exploratory design problems.

Most machine methods for exploratory design consist essentially of isolating the functions of technical systems and their elements and finding means for implementing these functions. In this connection, a methodology is developed and presented (Chapter 4) for analyzing the functions of technical systems and the construction of their functional structures -- the foundation for synthesizing physical operating principles and technical treatments.

The book presents machine methods for exploratory design and their application for the basic three types of search problems singled out above: physical operating principles (Chapter 5), technical treatments for a given physical operating principle (Chapter 6,7), and optimal parameters of a given technical treatment (Chapter 9,10). The most effective and promising methods are those of synthesizing physical operating principles, as well as the direction of using mathematical programming approaches and methods to solve problems of finding new technical treatments (Chapter 8,9), (and, obviously physical operating principles), which is illustrated with the example of solving problems of synthesizing the optimal forms of technical system elements (Chapter 11).

Although there is no doubt about the advisability of using the individual methods, combined and integrated utilization of these methods is more effective in designing certain classes of technical systems. In this connection, the conception of creating exploratory design subsystems for automated design systems is presented (Chapter 12,13). These subsystems are aimed at solving all three types of exploratory design problems, and have the appropriate software, hardware and information support. Special attention is devoted to substantiating the efficiency of exploratory design subsystems (Chapter 14).

The results which have been achieved with regard to machine implementation of individual methods and using these methods as the basis for creating the first experimental exploratory design subsystem are now relatively primitive and inefficient, and if we consider the substantial amount of work on preliminary data preparation, they are not always competitive with traditional "manual" technology based on the trial and error method. Current attempts to automate exploratory design can be compared with the creation of the first artillery pieces, which were far inferior to the bow and arrow in terms of range, accuracy and rate of fire, as well as the first automobiles, which were significantly inferior to horse-drawn carriages and wagons in terms of capacity, speed and drivability.

In this connection, the main result today consists of recognizing the requirement for automating exploratory design, providing theoretical and experimental proof of the possibility of solving this problem and providing a substantiated statement of practically useful goals and tasks. While it took several hundred years in the Middle Ages to convert primitive artillery pieces to efficient weapons, and it took several decades for the automobile to develop in the late 19th and early 20th centuries, if we consider the increasing rate of technical progress, we can assert that the road from today's primitive exploratory design systems to efficient systems in the last quarter of the 20th century will be covered in 10 or 15 years.

Let us attempt to formulate the basic directions of work which will facilitate the final formation and extensive practical application of machine technology for exploratory design. First of all, the approaches and methods for solving individual problems in the area of automating exploratory design which are considered in the book need to be improved (and even replaced with principally new, more efficient ones). In addition, we note the most important new directions of theoretical and experimental research and practical development.

1. The practical application of individual methods and systems (or subsystems) for exploratory design involves extremely laborious work to create data bases, i.e., to prepare information and transfer it to machine media. This is especially clear, for example, in constructing a general AND--OR tree of technical treatments and then maintaining and updating it. In this connection, one of the main directions of the work consists of automating data preparation, or more precisely creating methods, algorithms and programs which reduce significantly the amount of labor required.

2. The quality of the input data is very important in creating data bases. In addition, keeping in mind the fact that many information files are to a great extent invariant and will be used repeatedly in different areas of technology, it becomes necessary to develop standard forms for describing information for exploratory design systems. We have in mind the description of patent information, scientific discoveries, new physical (including chemical and biological) effects and laws, as well as new substances and construction materials, construction elements, etc. The standard forms for describing various units of information should basically be filled out by the authors who create the information, which will ensure high quality and facilitate essentially "automatic" development of data bases.

3. There now exist methods for automated solution of individual types of problems: 1) synthesis of new operating principles; 2) synthesis of new technical treatments for a given operating principle with the availability of several known technical treatments; 3) determination of optimal parameters of technical treatments. The following important problems, which facilitate complete automation of exploratory design, remain unresolved in the sense of efficient computer application:

-- using new physical operating principles (for which there exists no developed technical treatment) as the basis for designing acceptable technical treatments;

-- statement of problems of optimizing parameters, and selecting a method of solution, for new technical treatments.

4. The basic body of information in the technical sciences consists of descriptions of fairly precise mathematical models (for calculating and analyzing certain technical treatments), and their theoretical and experimental basis. New physical operating principles and new technical treatments have been evaluated intuitively, and occasionally experimentally, in engineering practice. In connection with automation of exploratory design, it has become necessary to develop methods of mathematical modeling for evaluating new physical operating

principles and technical treatments. The precision of these methods must facilitate the proper selection of only the best of several physical operating principles (or several technical treatments), i.e., it may be significantly less precise than the traditional mathematical models used to determine the numerical values of the parameters of planned technical systems.

5. Now that the first experimental autonomous exploratory design systems have been created (which have small data files, severely limited sets of solvable problems and small configurations of software and hardware), the next problem is to create the first practically useful efficient exploratory design subsystems which are organically interfaced with automated design systems, i.e., the problem of creating a second-generation automated design system [50]. Besides eliminating the shortcomings inherent in the first experimental systems, these subsystems should also devote special attention to service software, which makes it possible for designers to master a new tool quickly and easily.

6. Preliminary analysis has indicated that more than half of the software and information and methodical support used in exploratory design subsystems is invariant to the class of technical systems being designed. In this connection, the important problem arises of creating standard software and information and methodical support for exploratory design subsystems.

These directions reflect the current level of knowledge and understanding of the problem in question, and naturally do not pretend to be exhaustive even allowing for other directions and problems formulated in the book in the course of presenting individual problems. However, regardless of the limited nature of this knowledge, the primary determining direction has become obvious: reinforcing human intellect with the help of machines. The creation of automated systems for exploratory design and their integration with automated design systems, automated experimental and scientific research systems and information retrieval systems for patent and other scientific-technical information will make it possible to facilitate the highest possible rate of technical progress and to maintain the status of a technically developed country.

6900
CSO: 1863/63

DATA PROCESSING EQUIPMENT

Moscow TEKHNICHESKIYE SREDSTVA OBRABOTKI INFORMATSII in Russian 1981 (signed to press 17 Feb 81) pp 2, 319-320

[Annotation and table of contents of book "Data Processing Equipment", by Vasiliy Nikolayevich Kriushin, Nikolay Matveyevich Surin, Valeriy Pavlovich Chuprikov and Nina Grigor'yevna Chernyak, Izdatel'stvo "Finansy i statistika", 12,000 copies, 320 pages]

[Excerpts] This book has been authorized by the USSR Ministry of Higher and Secondary Specialized Education as a textbook for students at higher educational institutions who are studying in the specialization "Organization of Mechanized Processing of Economic Data."

Annotation

This textbook describes the operating characteristics, structural and schematic diagrams, and primary units of contemporary keyboard, keypunch, and small electronic computer machines. It presents the methodological foundations of selecting the set of hardware for automated control systems and organizing technical servicing of such systems. Considerable attention is devoted to procedures and methods of work on the machines and the methodology of programming solutions to economic problems. The book can be used by a broad range of practical workers in the field of the application of computer technology.

Table of Contents

11,176
CSO: 1863/51

HOMOGENEOUS COMPUTER SYSTEMS, STRUCTURES AND DEVICES

Moscow ODNORODNYYE VYCHISLITEL'NYYE SISTEMY, STRUKTURY I SREDY in Russian 1981 (signed to press 15 Jun 81) pp 2-11, 197-208

[Annotation, introduction, chapter 9 and table of contents from book "Homogeneous Computer Systems, Structures and Devices", by Eduard Vladimirovich Yevreinov, Izdatel'stvo "Radio i svyaz'", 10,000 copies, 208 pages]

[Extracts]

Annotation

This book presents the fundamentals of the construction of homogeneous computing systems, structures and devices which use a model of the body of computers which is based on principles of parallel execution of operations, variable logic structure, and design homogeneity of the elements and connections between them.

The book is intended for scientific workers and engineers specializing in the area of computer technology and cybernetics.

Sixty figures, 3 tables, 21 bibliographic references.

Reviewers: Professor S.D. Pashkeyev and Professor Yu. M. Shamayev.

Introduction

Upon the appearance of the computer, it was clear that computer technology can be used in many areas of human activity. Subsequent research and practical utilization of computers demonstrated that there is no branch of the national economy in which the use of computer technology would not provide significant economic effect . It became clear that, along with power engineering, computer technology predetermines the capabilities of society to increase labor productivity.

Increased labor productivity in a particular branch of the economy is determined, first of all, by continual expansion of the sphere of application of computers and, secondly, by constant growth in the complexity of the problems to be solved.

The requirement for computation has now increased to the extent that there is now a requirement for qualitatively new mass computing technology which combines high efficiency, reliability and computational economy with convenience and simplicity of utilization.

Mass computing technology and its technical and economic indicators. Mass computing technology has a great deal in common with service technology. It must provide users with complete computer support, low computational cost and high reliability, convenience and simplicity of utilization and servicing, etc. In other words, the computer must be just as simple and inexpensive as any other everyday item: television sets, tape recorders, etc. No special conditions or special training are required to accommodate and use the computer. The requirement for low-cost computation is foremost; providing a capability for augmenting computational resources as the complexity of the problems which are solved increases is also important.

The direct relationship between labor productivity and the amount of computing resources and the social requirement for continually increasing labor productivity lead to a requirement for continued increase in the amount of computer resources with overall cost limitations. Under these conditions, the creation of mass computing technology becomes one of the most urgent problems at the current stage of social development.

The specific computing capacity V_0 is the quantitative estimate of mass computing technology. By analogy with power availability, this indicator characterizes the availability to society of computing resources per individual. The specific computing capacity is determined by dividing the total capacity of all computing resources by the total population of the country, and is measured in the number of standard computational operations executed per unit of time per individual.

Another estimate of mass computing technology is the specific computing cost C_0. This indicator is determined by dividing the total capital expenditures required to create the country's computing resources by their total capacity.

The specific computing capacity and specific computing cost are the primary indicators which characterize the level of development of computing technology.

It is easy to understand that the annual expenditures for computing technology cannot exceed some defined portion of the national revenues (5-10%). Of course, with this limitation it is possible for V_0 to increase continuously only if there is a corresponding continuous drop in C_0.

Analysis of the trends of development of computer technology and requirements for computing resources to solve national economic problems in order to facilitate the required rates of increase in labor productivity allows us to conclude that the specific computing capacity V_0 must approximately double annually up to 1 billion operations per second per individual. Then V_0 can double every 10 years. Such an increase in V_0 will fully satisfy the requirements for computing resources, allowing for the rate of population increase in the country. This increase in V_0 can be provided if C_0 is reduced to 1 ruble per 100 operations/second during the first 10 years, to 1 ruble per

1000 operations/second in the second 10 years, and finally to 1 ruble per million operations/second (Figure I.1).

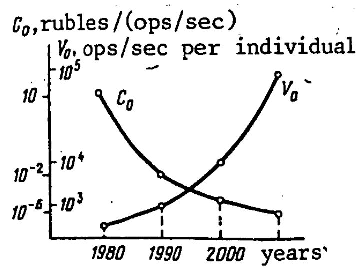


Figure I.1.

With V_0 and C_0 changing in this manner, three stages can be isolated in the development of mass computing technology. In the first stage, with $V_0 = 10^3$--10^4 ops/sec per person and C_0 = 0.01 rubles per operation per second, the required level of provision of mass computing technology is achieved and its cost reduced simultaneously by two or three orders of magnitude as compared with the cost of third-generation computing equipment. In the second stage, with $V_0 = 10^6$ ops/sec and C_0 = 0.001 ruble, the requirement for solving mass problems while further reducing the cost of computer facilities through mass production of same is satisfied. In the third stage, when $V_0 = 10^9$ ops/sec and C_0 = 0.000001 rubles, the requirement for solving complex problems while sharply reducing the cost of computing facilities by switching to a new production technology is satisfied.

Unlinked, weakly-linked and strongly-linked problems. All problems which can be solved using computers may be divided into three classes, depending upon the structural singularities of the links: unlinked, weakly-linked and strongly-linked.

Unlinked problems represent a set (aggregate) of independent problems, each of which can be solved within some accepted tolerable time using the resources of one computer taken separately. In the simplest case, a set can contain a single problem. If we assume that the acceptable problem solving time is 10^4 sec (approximately 3 hours), then the maximum complexity of a problem to be solved amounts to 10^{10} -- 10^{11} operations for third-generation machines operating at 10^6 -- 10^7 ops/sec. When the set includes problems with maximum complexity of 10^{11} operations and the problems are not linked, a solution can be achieved either by using the same computer serially, or by several computers operating in parallel.

Weakly-linked problems represent a collection of problems which are interrelated in terms of information exchange such that the total volume of interactions for the entire set of problems does not exceed the volume of computation for a single problem. Obviously, weakly-linked problems cannot be solved directly within the accepted tolerable time, since their complexity may significantly exceed the capabilities of a single computer, and it may be difficult to use several computers because of the information links between the problems.

Furthermore, one important property of weakly-linked problems should be pointed out. Because of the small amount of linkage between problems, there are methods which can be used to transform a weakly-linked problem to a set of unlinked problems requiring a slightly larger amount of computation.

Strongly-linked problems represent a set of problems which are interrelated in terms of information exchange such that the total amount of interactions for the entire set is bounded by the amount of computation for one problem and the amount of computation for the entire set. It is impossible to solve such a problem on a single third-generation computer within the tolerable time, and the lack of methods for dividing a problem into independent parts makes it impossible to solve these problems using a set of independent machines as well.

Unlinked problems are mass problems. In order to satisfy the demand of the national economy for the solution of mass problems it is necessary to facilitate an increase in the overall performance of the country's computer pool; the performance of unified computing resources must be improved in order to provide for solving weakly-linked, and especially strongly-linked, problems. Since complex problems involving a large amount of computation (strongly-linked problems) are encountered significantly less often than simple (unlinked) problems, these two are close to one another in terms of the total amount of computation required. In this connection, it is useful to pose the problem of developing unified computing facilities which are identically convenient for solving both a large number of unlinked problems using separate computer resources, as well as a small number (within a single problem) of strongly-linked problems using generalized computing resources.

Model of computer collective. In order to solve all three classes of problems (unlinked, weakly- and strongly-linked) it is necessary to turn to a computational model which is qualitatively different from the traditional computer model.

A computer model is the product of formalizing the actions of a human computer engaged in problem solving according to predefined formal solution rules. The computer model is built on principles of serial execution of operations, fixed computational logic structure, and design heterogeneity of the basic sections of the model and connections between them. The computer model provided the basis for the construction of first-, second- and third-generation machines.

An essential shortcoming of the computer model is the presence of a theoretical limit for computational capacity resulting from the finite information transmission rate between elements of the model when operations are executed serially. As the theoretical limit is approached, the technical problems involved in creating a computer with the required parameters increase, and costs also go up sharply. As a result of the foregoing, regardless of a fairly high theoretical computational capacity limit (10^9 ops/sec), we must limit ourselves to a technical limit of 10^7 ops/sec for third-generation machines.

Another shortcoming of the computer model is the relatively large expenditures on software and hardware with limited performance, which results in a high specific computational cost. Even for the most sophisticated third-generation computers the specific cost amount to between 1 and 10 rubles per operation per second, and cannot be reduced significantly.

The computer collective model proposed by the author in the late 1950's and early 1960's for constructing efficient high performance computer systems differs in principle. This model is the result of formalizing the computational process executed by a group of computers in solving a unified complex problem. The complex problem is represented as a set of interrelated parts. Each part is solved by an individual computer, and information is exchanged when necessary with other computers which are then solving their own parts.

The computer collective model is based on principles of parallel execution of a large number of operations, variable logic structure and design homogeneity of the elements and interconnections.

When using this model, the computing equipment is represented as an aggregate of computer models combined into a unified computer collective model with the help of information exchange between one another.

In solving a complex problem, its component interrelated parts are distributed among the computers and solved in parallel. The structure of the information exchange network is established in accordance with the scheme of interactions between sub-problems. In solving a complex problem which represents an independent set of simple problems, the latter are distributed among computers for parallel processing. There is practically no exchange between computers.

The computer collective model has no theoretical performance limit thanks to the assumption of a theoretically unbounded increase in capacity by adding additional computers. The variable logic structure and design homogeneity make it possible to achieve high reliability indicators and economy for the computer collective model.

Computer facilities based on the computer collective model can be used to solve all three classes of problems. The problem is actually to mass-produce computers which are convenient and easy to use, reliable, economical, and suitable for combination into a computer collective model for solving complex problems requiring major unified computing resources. The possibility of creating a unified physical-technological base and organizing mass production of a small number of types of computers allows us to hope for an optimal solution to the problem of satisfying the demands of the national economy for computer technology.

The theoretically unbounded capability for increased capacity of computer devices based on the collective model while simultaneously facilitating conditions for mass production makes it possible to achieve the required specific computing capacity while keeping the required specific cost within the necessary limits in accordance with the demands of the national economy.

At the present stage of development of computer technology, a transition to creating computer facilities based on the computer collective model is becoming realistic thanks to the achievements of microelectronics in creating inexpensive, reliable, small processors and microcomputers. It should be noted that the computing resources of an individual microcomputer are limited. This amplifies further the trend toward constructing computer facilities on the basis of the computer collective model.

The difficulties in creating efficient software are also reduced significantly here as a result of constructing distributed means for controlling computational processes.

The changeover to computer facilities consisting of a large number of relatively simple computers of the same type with regular interconnections makes it possible to simplify significantly their development, manufacture, debugging and operation.

All of this points toward the advisability of switching over to mass computing facilities constructed on the basis of the computer collective model.

Homogeneous computing systems, structures and facilities. The technical foundation for implementing the computer collective model is the integrated direction of homogeneous computing systems, structures and facilities which was proposed and theoretically justified by the author in the late 1950's and early 1960's. Now, after more than 20 years' development of this direction, it has become obvious that it provides the basis for constructing mass computing facilities intended for solving unlinked, weakly- and strongly-linked problems.

This direction includes many classes of computing facilities: distributed computer systems, concentrated computer systems, homogeneous computer structures and homogeneous computing facilities.

Distributed computer systems represent an aggregate of spatially separated computers (or computer systems) which are interconnected via communications systems such that the required interchange between system elements is supported. A distributed computing system can operate in the computer network mode, in which a set of independent problems is solved, as well as in a dedicated mode in which a single complex problem is solved using the overall computing resources of the system. The limiting case of a distributed computer system is the unified distributed computer system, which combines all of the computer resources of the country and uses them to solve both individual simple problems as well as a single complex problem.

At the other pole we find homogeneous computing facilities, which represent a set of elementary computers which cannot be simplified any further (elementary automata with programmable structure) which are linked together with regular communications. Homogeneous computer facilities provide an ideal basis for constructing computer facilities in a continuous technological process. Thanks to the simplicity of the elements and the regularity of their connections, the computing facility is manufactured in the first stage in a unified technological process. In the second stage, each user uses software to set the facility to implement any universal or special-purpose computer which is maximally suited to the singularities of the problem at hand.

These two extreme classes of homogeneous computing systems encompass various computing systems and structures which differ in the dimensions and types of elements, and the communications configurations.

Homogeneous computer systems, structures and facilities now represent the most promising direction in computer technology: they make it possible to eliminate the limitations which third-generation machines cannot overcome with respect to providing high performance for solving complex problems while sharply reducing cost.

The changeover to industrial production of homogeneous computer systems, structures and facilities will make it possible to facilitate uniform distribution of problems within directions which have already been assimilated by industry, concentrating primary efforts in creating families of computers and minicomputers on user service quality, and on the operating efficiency of time-sharing computer systems in creating multiprocessor systems. Homogeneous computer systems, structures and facilities should be tasked to solve the problems of achieving high performance and low specific computational cost, as well as facilitating the transition to mass computing technology.

9. APPLICATION OF HOMOGENEOUS COMPUTER SYSTEMS, STRUCTURES AND FACILITIES FOR SOLVING COMPLEX PROBLEMS

9.1. Application of Homogeneous Systems for Solving Problems in Economics

The first chapter pointed out that homogeneous computer systems, structures and facilities are needed to solve all three classes of complex problems: independent, weakly- and strongly-linked. This actually means that homogeneous systems provide universal means for solving both mass problems involving relatively small amounts of computation, as well as individual complex problems whose solution requires the execution of a large number of operations. Since it is necessary to use computer technology in all areas of the national economy, and considering the increasing requirements for the execution of large amounts of computation, homogeneous systems become mass computing facilities.

Under these conditions, it is difficult to enumerate all of the areas in which homogeneous systems can be applied effectively in the national economy. Therefore, we shall consider some of the most important ones.

Once such area of homogeneous system application is solution of problems in economics, including problems involved in economic management. The difficulties in solving these problems stem from the increase in their complexity due to the continuing increase in the number of products produced, the increase in the number of component parts of products, the increase in the complexity of the technological processes involved in their manufacture, and the rapid obsolescense of products.

The labor intensity of management problems increase to an even greater extent. The complexity of many economic management problems is determined quantitatively by material flows. An intensive increase in the ties between enterprises is observed as the economy develops. We can conclude from this that the increased complexity of economic management problems does not depend so much upon the number of objects being managed as the number of connections between them. It has been shown on the basis of experimental investigations that the complexity of economic management problems increases faster than the square of the total number of people involved in the economy. As the economy develops, a moment will arrive at which the overall complexity of the management problems will exceed the human resources of the entire society which could be used to solve these problems.

Thus, if the limit of the economic management tasks which one person could solve was reached earlier, which made it necessary at that time to change over to collective management, we are now approaching the limit of the capabilities of the entire society to manage the economy without using technical means. The need for using computers to solve problems of managing the economy is clear from this. According to estimates made in the late 1960's, effective management of the economy in this country at that time required computing facilities with a capacity of over 300 million operations per second.

Unfortunately, the problem of managing the national economy cannot be split into independent problems which could be solved using several hundred isolated computers. This means that homogeneous computer systems which make it possible to solve complex problems with the required capacity are needed to solve the problems involved in managing the national economy.

Solving the problem of managing the economy requires involving hundreds of thousands of specialists, who are spread about the country, in the computational process. This means that successful solution of the problem of managing the economy requires that we switch over to creating distributed homogeneous computer systems which combine the principles of centralized management and distributed data processing in the proper proportions.

Thus, the creation of the National Computer Center Network and the construction of the Nationwide Automated Data Acquisition and Processing System for accounting and managing the national economy will become possible if homogeneous computer systems are widely distributed throughout all branches of the national economy and joined into a unified national distributed homogeneous computer system.

The central question in creating a unified distributed homogeneous computer system involves development of a rayon homogeneous computer system. This is because rayon systems are designed to solve problems which arise within administrative rayons, of which there are several thousand in the country. Therefore, rayon problem solving is of a mass nature. Hence, mass computer technology is required which must have both high performance and low cost. Based on the rayon population and the requirement for providing specific computing capacity per person of $V_0 = 10^4$ operations per second, the total capacity of the rayon homogeneous computer system must reach 10^9 operations per second with a specific computational cost per operation per second of $C_0 = 0.001$ rubles. The rayon homogeneous computer system encompasses about 1,000 subscribers (enterprises, kolkhozes, sovkhozes and institutions), with connections between them comprising a total of about 1,000 kilometers. The capacity of the communications channels between elementary machines in the rayon homogeneous system must be approximately 1 mbps. Under these conditions, the rayon homogeneous system will achieve a capacity of 10^9 operations per second regardless of the deployment of the elementary machines within the territory of the rayon.

For a rayon homogeneous system it is advisable to co-locate the computer facilities with the communications facilities; it is most convenient to install rayon homogeneous systems at communications centers, especially at automatic telephone exchanges. Then all subscribers have direct access to the rayon system using existing communications channels. The presence of the rayon homogeneous computer system at an automatic telephone exchange makes it possible to place many of the functions of automatic communications (all the way up to implementation of switching) on the computer facilities. Combining computer and communications facilities in a unified data transmission, storage and processing system makes it possible to form a unified technical and organizational base for solving problems involved in managing the economy with maximum efficiency.

In this respect, the rayon homogeneous computer system can be called the primary system at the rayon level, similar to the fashion in which the primary network is used to build up different communications systems. By combining all of the rayon homogeneous computer systems into a unified nationwide distributed computer system it is possible to obtain a primary data transmission, storage and processing system. The unified rayon homogeneous computer system can be used as the basis for the Unified System of Computer Systems and the Nationwide Automated Data Acquisition and Processing System for accounting and management of the national economy.

9.2. Application of Homogeneous Computer System for Solving Complex Scientific and Technical Problems

The current state of science and technology is characterized by the occurrence of a significant number of complex problems whose solution requires high-capacity computing facilities. There is an increase in the complexity of computation in certain classes of problems, and new classes of problems are also arising in which the solution complexity is even greater. Such problems include problems of computational mathematics, mathematical physics, linear and nonlinear programming, pattern recognition, investigation and design of complex systems, modeling of complex systems, etc. There is now a requirement for solving complex problems involving 10^{13}--10^{14} operations with 10^{10}--10^{11} bits of information to be stored. The solution of such problems requires computing facilities with capacities of 10^{9}--10^{10} operations per second.

Ordinary computers based on the computer model have a technical performance limit of 10^{7} operations per second. Under these conditions, complex problems can clearly be solved if we change over to homogeneous computer systems based on the computer collective model and having capacity of 10^{9} operations per second or more.

Concentrated homogeneous computer systems are most convenient for solving complex scientific and technical problems: these can be based on modern microcomputers, which are characterized by small size, low cost and high reliability.

Let us consider some classes of complex problems and singularities of utilizing concentrated homogeneous computer systems to solve them. Many problems in radar, hydroacoustics, nuclear physics, geophysics, meteorology, medicine, sociology, etc. can be solved using pattern recognition methods. However, high capacity computer systems are required to solve these problems; in addition, the requirements for efficient man-computer system information exchange make it necessary to develop methods for inputting and outputting optical and audio information using pattern recognition methods.

Pattern recognition problems are divided into three types. The first type involves finding a decision rule in which the expenditures for construction are minimal. The second involves finding a system of features with which the costs associated with recognition errors and with measuring these features are minimized. In problems of the third type it is necessary to find the version of grouping realizations into patterns which would minimize the expenses involved in using the patterns. Problems of empirical prediction are close to pattern recognition problems.

Empirical prediction problems involve developing methods of representing initial data, detecting regularities in a set of input data and representing algorithms for predicting new regularities. The problem of empirical prediction is one of the main parts of the problem of artificial intelligence.

The solution of pattern recognition and empirical prediction problems involves a large amount of computation. Furthermore, these problems can be represented as an aggregate of interconnected problems of lesser complexity, which makes it relatively easy to construct parallel algorithms which can be implemented efficiently in concentrated homogeneous computer systems.

An important class of problems whose solution is of major interest is comprised by problems involved in automating the design of complex articles and modeling complex systems. The problem of reducing planning periods and putting articles into production is now becoming especially critical in machine building. This problem can be solved by creating automated systems for planning, technological preparation and management of production on the basis of homogeneous computer systems. One of the planning problems is to develop effective methods of constructing the geometry of the planned object. It should be noted here that this problem is particularly complex in constructing the flow surfaces of flight vehicles, ship hulls and light truck cabs, since the design requirements call for optimal equipment placement and strength while the outside surface must be smooth in accordance with the requirements of hydroaerodynamics and aesthetics.

When machine methods are used to solve planning problems, problems arise in developing the mathematical description of complex contours and surfaces, methods of configuring parts into assemblies, and determining the differential and integral characteristics of the surface. These problems can be solved especially effectively using the theory of nonlinear and linear cubic splines. Spline-function methods have made it possible to create an effective mathematical apparatus for representing contours and three-dimensional surfaces which is universal, homogeneous and easily reduced to algorithmic form. The particular value of these methods is their convenience in implementing parallel computational processes in homogeneous computer systems.

An important area of application of homogeneous computer systems is the investigation of complex object such as microcosmic physical systems, biological and linguistic systems, etc. In terms of properties, all of these objects are complex systems whose main singularity is that they cannot be divided into independent parts. This makes it necessary to examine the entire system as a unified whole. It becomes possible to solve such problems if concentrated homogeneous computer systems are used.

9.3. Application of Homogeneous Computer Systems for Implementing Structural Models

Constructing a homogeneous computer system on the basis of the computer collective model makes possible a different approach to the solution of a number of complex problems. The use of homogeneous computer systems is especially promising for the construction of digital models. Modeling devices have proven themselves in analog technology; their advantages are that it is simpler to assign the problem solving method, and the solution is accelerated, thanks to the structural representation of the problem in the model. The simplicity of assigning the problem solving method is combined with simplicity in changing the solution scheme during the modeling process. Structural representation of a problem makes it possible to observe the behavior of sections of the model as a function of parameter variations.

Homogeneous computer systems can be used to design various automation devices, control systems, communications devices, special-purpose machines and automata. An important circumstance in structural modeling is the capability of evaluating the implementation complexity of a device or machine. Using homogeneous computer systems, we can assign structural models of widely varying special-purpose machines and devices without spending a lot of time. This makes it possible to find the optimal structure in accordance with given criteria by trial and error. The modeling system can be represented as the combination of a homogeneous computer system and computer structure or device. A modeling system based on a homogeneous computer system has all of the advantages of analog systems in terms of convenience and problem-solving simplicity, but avoids the shortcomings inherent in analog devices, namely low computational accuracy and the requirement of using manual methods for structure assignment.

Let us consider some examples of implementing the computer collective model in communications devices. One of the first examples of using digital modeling systems in communications engineering is the microprocessor-based homogeneous control system for existing electromechanical and quasi-electronic switching systems. Such a system makes it possible to expand the scope and type of services available to subscribers and to improve the technical servicing of existing electromechanical systems; it also facilitates a gradual transition to purely electronic systems and the associated possibility of gradual training and retraining of personnel.

To these advantages of using microprocessor-based homogeneous control systems we can also add increased reliability, which provides high viability for homogeneous systems, and relatively low cost of control devices, which facilitates mass production of small-nomenclature devices, good hardware repairability and the capability to increase the number of switching fields serviced easily by decentralizing control and adding control modules.

Automatic telephone exchange controllers using the computer collective model are a natural complement to existing switching systems, since a switching field is nothing more than a certain collective of connectors. In addition, this collective has the same intrinsic functions as a computer collective: connections are made by homogeneous devices (homogeneity), several connections can be made simultaneously (parallel operation); depending upon the number of subscribers serviced, exchange switching systems are built up by combining the necessary number of standard switching field modules. This latter property is analogous to the variable logic structure in the computer collective model. By establishing a one-to-one correspondence between the connector collective model and the computer collective model, we can thus realize automatic telephone exchanges based on these principles.

Figure 9.1 shows a functional diagram which illustrates the design principle of an electromechanical switching system with a ring-type microprocessor-based homogeneous control system. The switching equipment establishes the connections directly. The service equipment provides the necessary signal interface between the switching equipment and the microprocessor-based control system. One characteristic feature of the control system, and incidentally of all homogeneous systems, is that the control system memory is made up of the memory of the individual microprocessors which comprise the system.

This control arrangement makes it possible to divide the switching field into separate zones served by separate machines in the computer collective model. When any control system microprocessor malfunctions, its functions are taken over by another processor which is free at that time, and the system continues to function.

Further development of this principle consists of dividing the switching field and controller into separate functional modules and constructing automatic telephone exchanges on the basis of the computer collective model which is realized using the modules. In the present version (Figure 9.2) the telephone exchange architecture is represented by a decentralized homogeneous system.

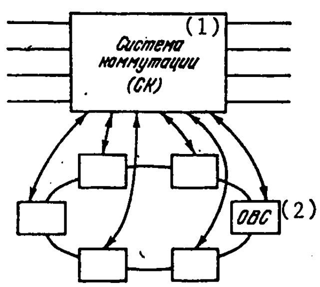


Figure 9.1.

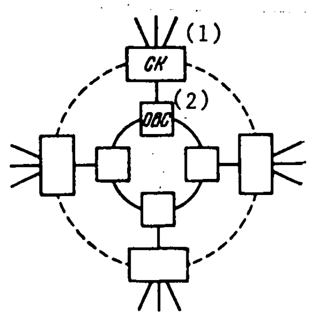


Figure 9.2.

Key:

1. Switching system;
2. Homogeneous computer system.

1. Switching system
2. Homogeneous computer system

The exchange consists of identical separate modules. A module is made up of a switch, which makes connections directly, and a microprocessor-based homogeneous control system, which organizes the process by which connections are made. In turn, the microprocessor-based control system consists of a collection of microprocessors which are combined into either a ring or a matrix system.

By isolating the required number of microprocessors, it is possible to provide the required capacity and memory size in this case both for making connections and for making various additional services available to subscribers, providing the required viability, system repairability, etc.

The microprocessor-based control systems of the individual modules are joined in the same manner, and represent the overall controller for the entire automatic telephone exchange. The modules are connected through both the required physical connection and under the control of the overall program which controls the individual modules in the system.

In a special case, the microprocessor-based control system of a single module can degenerate into a single microprocessor, but these microprocessors are nonetheless combined into the homogeneous control system of the exchange.

BIBLIOGRAPHY

1. Akushskiy, I.Ya. Yuditskiy, D.I. "Mashinnaya arifmetika v ostatochnykh klassakh" [Machine arithmetic in residue classes]. Moscow, Izdatel'stvo "Sovetskoye Radio", 1968.

2. Anisimov, B.V., Chetverikov, V.N. "Osnovy teorii i proyektirovaniya tsifrovykh vychislitel'nykh mashin" [Fundamentals of theory and design of digital computers]. Moscow, Izdatel'stvo "Vysshaya Shkola", 1970.

3. Balashov, Ye.P., Knol', A.I. "Mnogofunktsional'nye zapominayushchiye ustroystva" [Multifunctional memory devices]. Leningrad, Izdatel'stvo "Energiya", 1972.

4. Glushkov, V.M. "Sintez tsifrovykh avtomatov" [Synthesis of digital automata]. Moscow, Izdatel'stvo "Fizmatigiz", 1962.

5. Varshavskiy, V.I., et. al. "Odnorodnye struktury" [Homogeneous structures]. Moscow, Izdatel'stvo "Energiya", 1973.

6. Golubev-Novozhilov, Yu.S. "Mnogomashinnye kompleksy vychislitel'nykh sredstv" [Multi-machine computer complexes]. Moscow, Izdatel'stvo "Sovetskoye Radio", 1967.

7. Yevreinov, E.V., Kosarev, Yu.G. "Vozmozhnosti postroyeniya vychislitel'nykh sistem vysokoy proizvoditel'nosti" [Possibility of designing high performance computer systems]. Novosibirsk, Izdatel'stvo SO AN SSSR, 1962.

8. Yevreinov, E.V., Kosarev, Yu.G. "Odnorodnye vychislitel'nye sistemy vysokoy proizvoditel'nosti" [High performance homogeneous computer systems]. Novosibirsk, Izdatel'stvo "Nauka", 1966.

9. Yevreinov, E.V., Prangishvili, I.V. "Tsifrovoye avtomaty s nastranvayemoy strukturoy (Odnorodnye sredy)" [Digital automata with adjustable structure (Homogeneous media)]. Moscow, Izdatel'stvo "Energiya", 1971.

10. Yevreinov, E.V., Khoroshevskiy, V.G. "Odnorodnye vychislitel'nye sistemy" [Homogeneous computer systems]. Novosivirsk, Izdatel'stvo "Nauka", 1978.

11. Kalyayev, A.V. "Odnorodnye kommutatsionnye registrovye struktury" [Homogeneous switching register structures]. Moscow, Izdatel'stvo, "Sovetskoye Radio", 1978.

12. Lazarev, V.G., Savvin, G.G. "Seti svyazi, upravleniye i kommutatsiya" [Communications networks, control and switching]. Moscow, Izdatel'stvo, "Svyaz'", 1973.

13. Pashkeyev, S.D. "Osnovy mul'tiprogrammirovaniya dlya spetsializirovannykh vychislitel'nykh sistem" [Fundamentals of multiprogramming for special-purpose computer systems]. Moscow, Izdatel'stvo, "Sovetskoye Radio", 1972.

14. Pospelov, D.A. "Vvedeniye v teoriyu vychislitel'nykh sistem" [Introduction to theory of computer systems]. Moscow, Izdatel'stvo "Sovetskoye Radio", 1972.

15. Pukhov, G.Ye., Samoylov, V.D., Aristvo, V.V. "Avtomatizirovannye analogo-tsifrovye ustroystva modelirovaniya" [Automated analog-digital modeling devices]. Kiev, Izdatel'stvo "Tekhnika", 1974.

16. Balashov, Ye.P., Smolov, V.B., Petrov, G.A., Puzankov, D.V. "Mnogo-funktsional'nye regulyarnye vychislitel'nye struktury" [Multifunctional regular computing structures]. Moscow, Izdatel'stvo "Sovetskoye Radio", 1978.

17. "Mul'tiprotsessornye vychislitel'nye sistemy" [Multiprocessor computer systems]. Ya.P. Khegaturov, editor. Moscow, Izdatel'stvo "Energiya", 1971.

18. Prangishvili, I.V., et. al. "Odnorodnye mikroelektronnye assotsiativnye protsessory" [Homogeneous microelectronic associative processors]. Moscow, Izdatel'stvo "Sovetskoye Radio", 1973.

19. Prangishvili, I.V., et. al. "Mikroelektronika i odnorodnye struktury dlya postroyeniya logicheskikh i vychislitel'nykh ustroystv" [Microelectronics and homogeneous structures for constructing logical and computer devices]. Moscow, Izdatel'stvo "Nauka", 1967.

20. Glushkov, V.M., Myasnikov, V.A., Ignat'yev, M.B., Torgashev, V.A. "Rekursivnye vychislitel'nye mashiny" [Recursive computers]. Moscow, 1977 (ITMVT AN SSSR Preprint)

21. Marchuk, G.I., Kotov, V.Ye. "Modul'naya asinkhronnaya razvivayemaya sistema. Kontseptsiya" [Modular asynchronous developmental system. Conception]. In two parts. Novosibirsk, 1978. (VTs SO AN SSSR Preprint).

Table of Contents

6900
CSO: 1863/61

- END -